SF 298 MASTER COPY KEEP THIS COPY FOR REPRODUCTION PURPOSES

AFRL-SR-BL-TR-98-
0310

REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 24, 1998

3. REPORT TYPE AND DATES COVERED
Final - July 1, 1995 to Dec. 31, 1997

4. TITLE AND SUBTITLE
Robust Adaptive Algorithms for Reconfigurable Flight Control Systems

5. FUNDING NUMBERS
F49620-95-1-0341

6. AUTHOR(S)
Marc Bodson

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)
University of Utah
Salt Lake City, UT 84112

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research/NM
Bolling AFB, DC 20332-6448

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

19980410 055

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The objective of the project was to develop adaptive algorithms for application to reconfigurable flight control. An emphasis of the research was on robustness issues related to unmodelled dynamics and actuator saturation. Areas of investigation included: (1) the development of real-time parameter identification algorithms for situations where the information content of the data varies significantly over time; (2) the incorporation of integral compensation in adaptive control algorithms with bias cancellation capabilities; (3) the design of command limiting methods for the alleviation of problems associated with actuator saturation. The results showed that relatively simple multivariable adaptive control algorithms could be designed that were effective in providing flight control reconfiguration over a wide range of flight conditions. A detailed fighter aircraft model was used to evaluate the performance of the reconfigurable control laws in simulations.

14. SUBJECT TERMS
Flight Control, Reconfigurable Control, Adaptive Control

15. NUMBER IF PAGES
88

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Enclosure 1

Final Report:
Robust Adaptive Algorithms for Reconfigurable Flight Control Systems [1]

Principal Investigator:
Marc Bodson
Department of Electrical Engineering
University of Utah, Salt Lake City, UT 84112
(801) 581 8590 bodson@ee.utah.edu

1. Motivation and Objectives

Reconfigurable flight control systems provide significant advantages over conventional control systems, in particular increased probabilities of survival after failures and combat damage. Consider for example the picture of a fighter aircraft shown in Fig. 1. The shaded areas show portions of the aircraft that might be missing after a hypothetical damage.

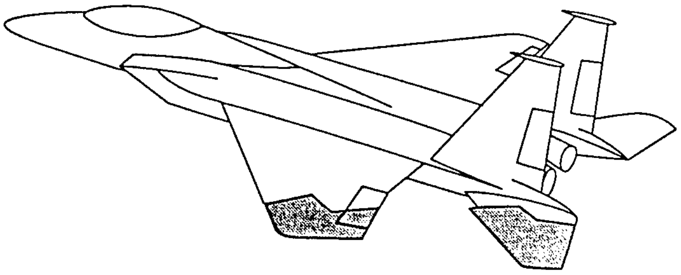


Figure 1: F-15 Aircraft with Hypothetical Damage

Such an incident would have a number of effects on the dynamic response of the aircraft, including: (1) a loss of the balance of the forces and moments acting on the aircraft, resulting in a divergent motion if the remaining control surfaces are not repositioned; (2) a loss of the stabilization effect of the left stabilator, yielding an aircraft with lower stability margins; (3) a reduction of the ability of

[1] Effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-95-1-0341. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation. The views and conclusions contained in the report are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

the control surfaces to generate pitching and rolling moments; (4) a loss of symmetry of the aircraft, so that a symmetric deflection of the stabilators produces a significant rolling moment, in addition to the pitching moment. If the control law is fixed and optimized for the unfailed aircraft, the damage may make it difficult to handle the aircraft, if not impossible. In contrast, a reconfigurable control system may be designed to restore the stability and the control of the aircraft by learning about its dynamic characteristics in real-time. As a result, steady flight may be maintained and tracking of the pilot commands restored.

While earlier approaches to control reconfiguration required the failure to be detected and identified, the objective of this project was to develop adaptive algorithms that did not require failure identification, so that reconfiguration could be achieved for a large number of failures and damages (including some which may not have been predicted at the design stage). A specific emphasis of the project was the study of robustness issues related to unmodelled dynamics and actuator limits. The results of the project have been reported in papers listed in the references. An M.S. thesis was also completed, which is included in Appendix. This report is an executive summary of the results found in the references.

2. Executive Summary of Research Accomplishments

2.1 Multivariable Adaptive Control Algorithms

Multivariable model reference adaptive control algorithms were developed for the application to reconfigurable flight control. The specific algorithms chosen were considered attractive because of their simplicity, which made them easy to implement in real-time, and useful in applications with limited computational resources. The performance of the algorithms was evaluated in simulations, using a highly detailed model of a jet aircraft, similar to the F-15 aircraft shown in Fig. 1. The model included a nonlinear aerodynamic data base, models of actuator dynamics with position and rate limits, engine dynamics, aerodynamic forces, and atmospheric properties. Trim changes and cross-couplings between the longitudinal and lateral motions, both significant effects of failures, were represented in the model.

The ability of the control laws to maintain steady flight and to restore tracking of pilot commands after unpredicted failures was demonstrated in [1], and their performance over a wide range of flight conditions was improved during the course of this project [2]. Fig. 2 shows some typical simulation results. The plots show the responses of the aircraft to a pitch rate command. The command consists in steps of 10 deg/s, 0 and -10 deg/s. The damage occurs 5 seconds into the simulation and consists in a missing left stabilator control surface. On the plots, the solid lines are the responses obtained with the adaptive control law, and the dashed lines are the responses obtained without reconfiguration, that is, with the parameters of the adaptive control law frozen at the time of the failure.

The plots on the first row show the pitch rate response (left) and the roll rate response (right)

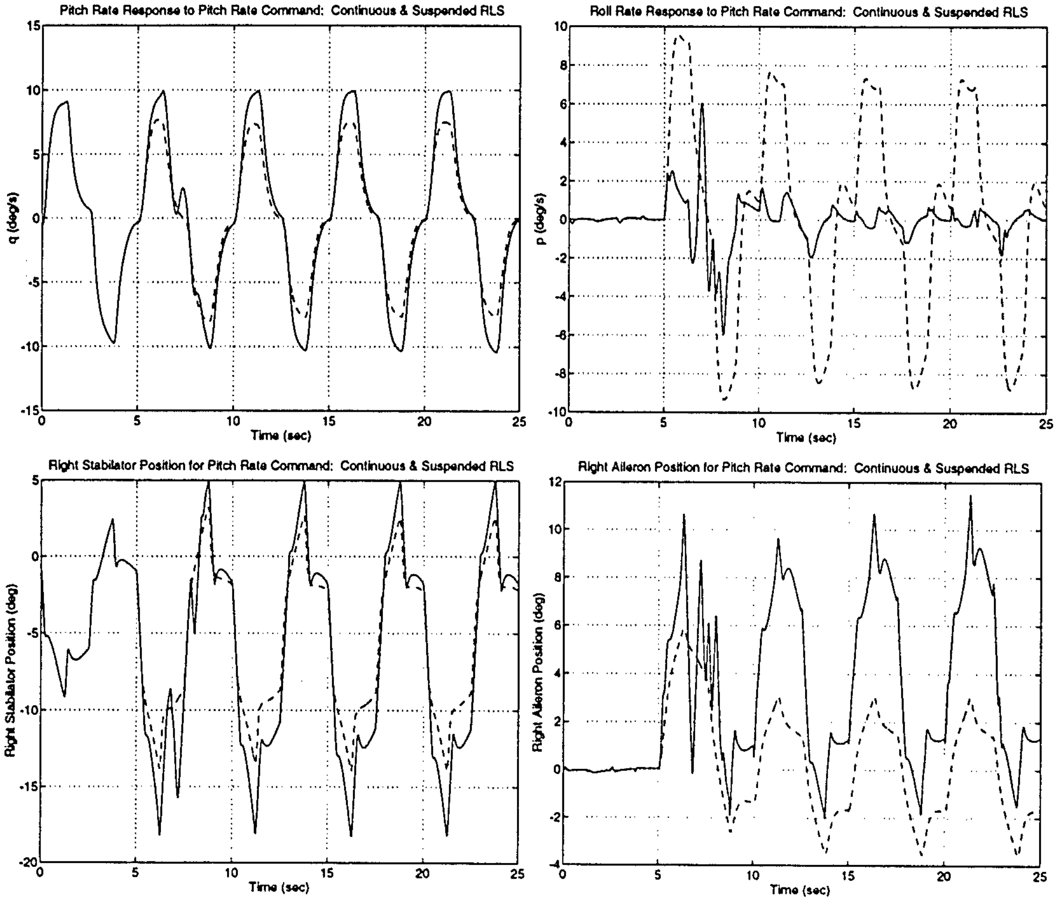


Figure 2: Responses with and without Adaptation after a Missing Stabilator

(both to a pitch rate command). Without reconfiguration, the desired pitch rate is not achieved because of the loss of pitching moment, and the roll rate is very large because of the coupling. With reconfiguration, the required pitch rate is achieved, and the coupling in roll rate is much reduced. The plots on the second row show the command to the stabilators (left) and the command to the ailerons (right). The stabilator command is doubled after the failure, as is to be expected since only one the stabilators is working. The aileron command, which was zero before the failure, becomes much larger after the failure in order to compensate for the rolling moment produced by the single stabilator.

The plots on Fig. 3 show the estimates of two significant parameters used by the adaptive control law. The parameter on the left is the amount of pitching acceleration generated by a unit deflection of the stabilators. The parameter is negative and decreases in magnitude by a factor of 2 after the damage, as expected. The parameter on the right is the amount of pitching acceleration generated by a unit change in angle of attack and is indicative of the inherent stability of the aircraft. A large negative value indicates a strong tendency of the aircraft to return to level flight after a disturbance.

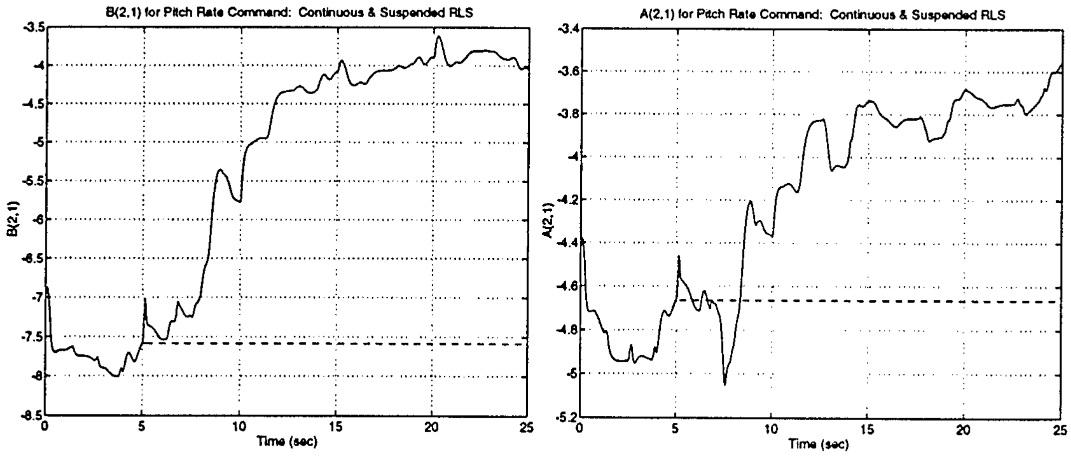


Figure 3: Adaptive Parameter Responses after a Missing Stabilator

The magnitude of the estimate is reduced after the damage, as expected because of the loss of the stabilization effect of the stabilator. Note that the parameters converge within 15 seconds, although the values that are reached within a much shorter time are sufficient to considerably improve the responses. Overall, the responses of Figs. 2 and 3 show that the adaptive system is able to correctly identify and compensate for changes in system dynamics caused by the damage.

2.2 Parameter Identification

At the core of the reconfigurable control law discussed in section 2.1 was a parameter identification algorithm capable of determining the necessary aircraft parameters. The problem of parameter identification for reconfigurable flight control is similar to the off-line identification problem, but involves additional difficulties which may be categorized as follows:

- *Lack of control over the actuator signals.* A significant difficulty arises because the actuator signals are determined by the control law and cannot be freely selected. Considerable improvements in identification performance can be obtained through optimization of the signals applied to the control surfaces. Conversely, in a reconfigurable control application, performance may be poor because the signals exhibit highly undesirable characteristics, including:

    1. high levels of correlation between the control signals and the aircraft states (in particular for linear state feedback control laws);
    2. long periods of quiescence (*e.g.*, in steady flight);
    3. highly coupled longitudinal and lateral motions (in particular for rapid roll maneuvers);
    4. excitation of nonlinear dynamics (for example inertial couplings).

In addition, because the control signals are determined by a control law which, itself, depends on the identification results, adverse interactions are sometimes induced between the identification and the control components of the system, further aggravating the problem.

- *Increased number of parameters for failed aircrafts.* Aircrafts often loose their symmetry after failures, requiring that more parameters be identified. For example, the roll effectiveness of a symmetric tail deflection may be assumed to be zero for an unfailed aircraft, but not otherwise. In general, accounting for possible failures may prevent the ganging of some surfaces, such as spoilers, and result in a significant increase in the number of parameters to be identified.

- *Real-time operation.* The computational requirements of a parameter identification scheme for reconfigurable control must be compatible with available computers. Further, the procedure must be performed with minimal supervision and may not require extensive trial and error adjustments by the pilot.

An interesting adaptive algorithm was proposed by Ward & Barron [2] to identify aircraft parameters in the face of these challenges. Their so-called modified least-squares algorithm incorporated a penalty on parameter variations in the optimization criterion. Ward & Barron observed that, without the penalty, an unacceptable trade-off was encountered in the design of a reconfigurable flight control system: either the responses of the adaptive algorithm were too slow during periods of significant excitation, or too noisy during periods of low excitation.

A recursive form of the algorithm was developed and analyzed by the principal investigator [3]. An averaging analysis showed that the algorithm exhibited some very interesting properties, including: (a) a variable data forgetting capability, so that the memory of the algorithm was long when the excitation was poor, and short when the excitation was significant; (b) a second-order filtering capability, as opposed to the first-order filtering of the original algorithm, reducing the rate of fluctuations of the parameters; (c) a stabilizing effect on the covariance matrix update.

Fig. 4 shows the results of a simulation with a single parameter system, illustrating the two features mentioned above. On the left is the response of the modified algorithm. On the right is the response of the standard least-squares algorithm with forgetting factor. The true parameter varies between 1 and -1 and appears as a triple step function. The smooth curves are the responses predicted by the averaging analysis, while the noisy curves are the responses of the parameter estimate. The approximate responses provided by the averaging analysis are found to be close to the original responses, validating the analysis. At time 400, there is a drop in excitation (and therefore of the signal-to-noise ratio). The response of the modified algorithm is seen to slow down after that, while the parameter fluctuations remain reasonable. For the standard least-squares with (the same) forgetting factor, it is found that when the excitation drops, the fluctuations in the parameters become very large. These fluctuations can be reduced by using a different

[2] D.G. Ward & R.L. Barron, *Self-Designing Flight Control Using Modified Sequential Least-Squares Parameter Estimation and Optimal Receding Horizon Control Laws,* Technical Report, Barron Assoc., Charlottesville, VA, March 1994.

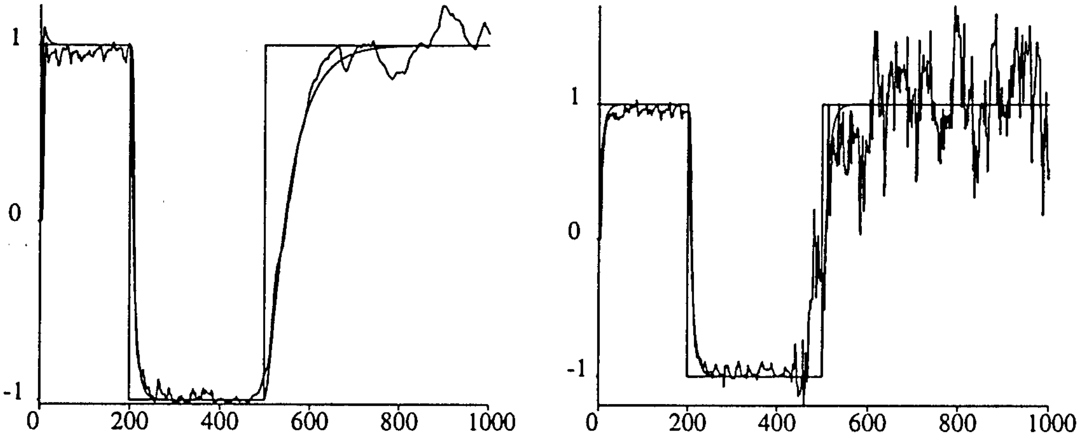


Figure 4: Adaptive Parameter Responses – Left: Modified LS; Right: Original LS

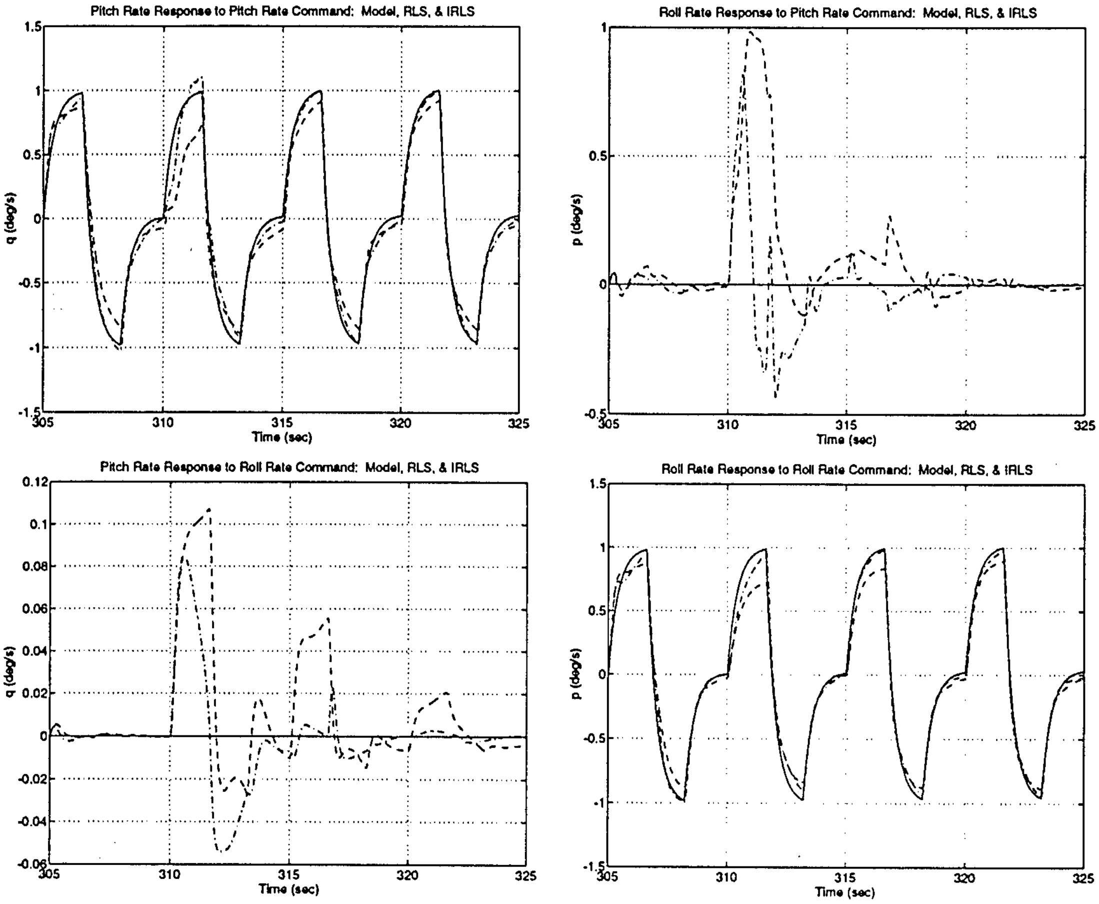
forgetting factor, but the response in the first part of the simulation becomes slower then. With the modified algorithm, the algorithm reacts fast when there is sufficient excitation, yet keeps parameter fluctuations reasonable when the information available cannot sustain a fast adaptation.

The identification algorithm was used in the simulations discussed in section 2.1, as well as in experiments. During the summer of 1996, a series of flight tests were performed with the VISTA F-16 aircraft, using the modified least-squares algorithm and a reconfigurable control law. These experiments were carried out by *Barron Associates, Inc.* under contract to the *Air Force,* and were not part of this project. However, the principal investigator assisted in the planning of the experiments and in the analysis of the data. The flight tests culminated in a successful landing of the aircraft under a simulated missing elevon, with an adaptive system reconfiguring rapidly after single or multiple impairments. The design of the parameter identification procedure as well as the experimental results, were reported in [4].

The adaptive algorithm was also tested experimentally under more modest circumstances in the principal investigator laboratory. In this case, the control law was a control law designed to avoid the excitation of resonant modes in a flexible beam. Such problem is more relevant to the control of flexible modes in aircraft structures than to the flight control problem. However, the experiments demonstrated again the effectiveness of the identification algorithm in a practical environment. The results were reported in [5], [6].

2.3 Integral Compensation in Adaptive Control

The incorporation of integral compensation in the reconfigurable control laws was considered, with the objective of improving performance, and of determining whether certain problems might be encountered. Indeed, it was found that the combined use of integral compensation and adaptive



Pitch Rate Response to Pitch Rate Command: Model, RLS, & IRLS

Roll Rate Response to Pitch Rate Command: Model, RLS, & IRLS

Pitch Rate Response to Roll Rate Command: Model, RLS, & IRLS

Roll Rate Response to Roll Rate Command: Model, RLS, & IRLS

Figure 5: Integral Compensation

bias cancellation could sometimes result in algorithms with uncontrollable modes [7]. Surprisingly, different conclusions were reached depending on the type of algorithm used. Specifically, the algorithms studied in this project (as well as those implemented in the Vista F-16 flight tests) were found to belong to a class of algorithms for which integral compensation and adaptive bias cancellation could be combined without uncontrollability problems. Given that integral compensation could be included in the model reference control algorithms, a question was then whether significant benefits could be derived from it. From extensive simulations, it was found that there was an advantage to the use of integral compensation, but that the improvement was less than might be expected from conventional control systems. To illustrate this point, Fig. 5 shows the results of simulations for control laws with and without integral action. The reference input was a sequence of steps in pitch rate (first row) and roll rate (second row). On the left are the pitch rate responses, and on the right are the roll rate responses. The solid lines are the desired responses, the dashed lines are the responses obtained without integral action, and the dot-dashed lines are those obtained with integral action. A failure occurred at t=310s, which consisted in a locked stabilator.

Overall, tracking of the commands is seen to be better when integral compensation is used.

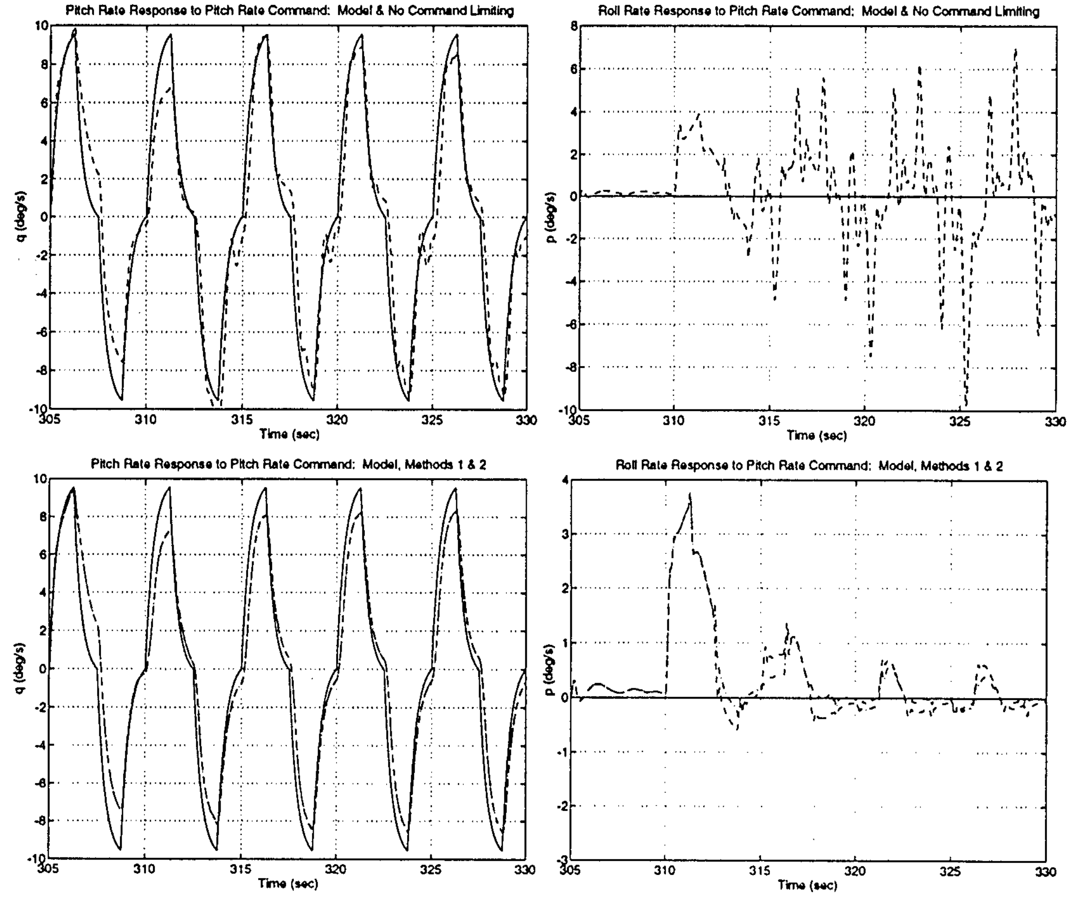
However, tracking is good even without the integral compensation. Trimming of the aircraft is also achieved without integral action (there is no visible drift in the responses). An explanation, provided in [7], is that the adaptive algorithm is able to fine-tune the estimates of the gain matrices and of the bias term in order to provide a compensation equivalent to that of integral control. Overall, it was concluded that, with a properly designed adaptive algorithm, integral compensation could be incorporated without uncontrollability problems, but also without benefits that were necessarily significant enough to warrant its use. While integral compensation is often included in the pitch axis of non-adaptive flight control systems, its purpose is mainly to assist with the trimming of the aircraft, so that an adaptive control law with trim bias estimation may not need integral compensation.

2.4 Actuator Saturation

Actuator limits pose a major problem in flight control system design. The positions of control surfaces are limited both in their range and in their rate of motion. If not accounted for in the design, these constraints may lead in flight to a significant degradation of performance and, sometimes, to pilot-induced-oscillations. The origin of the problems is in the delay of the response of the aircraft resulting from the actuator saturation as well as in the multivariable nature of the control problem, which is such that saturation in one axis may lead to poor responses in other axes.

For reconfigurable flight control systems, actuator saturation becomes even more problematic. Not only is control power reduced, but the couplings between longitudinal and lateral axes also become significantly larger after most failures. Simulations of the early model reference control algorithms yielded results that deteriorated significantly for large commands. The first row of Fig. 6 shows responses exhibiting this problem. On the left is the pitch rate response to a pitch rate command, and on the right is the roll rate response to a pitch rate command. The solid line is the desired response and the dashed line is the actual response. In the simulations, the left horizontal tail surface is locked at time t = 310 sec. The responses show a very large roll rate response to the pitch rate command. The large coupling is not observed for smaller commands of 1 deg/s and its appearance is due to the fact that the various actuators saturate independently, and in a way that prevents decoupling using a linear controller. The pitch rate response also exhibits glitches, which may be related to the sharp roll rate transients.

Four methods for command limiting were studied in [8] for their application to reconfigurable flight control. Significantly different concepts were considered: the scaling of the control inputs, the relaxation of the control requirements, the scaling of the reference inputs, and the least-squares approximation of the commanded accelerations. On the second row of Fig. 6, responses are shown which were obtained with the first two methods of command limiting. Plots for the other two methods were not very different and are not shown. One sees that tracking and decoupling performance is much improved with the methods of command limiting. All the methods gave comparable results, which was surprising since some methods were much simpler than others, and some methods used


Pitch Rate Response to Pitch Rate Command: Model & No Command Limiting

Roll Rate Response to Pitch Rate Command: Model & No Command Limiting

Pitch Rate Response to Pitch Rate Command: Model, Methods 1 & 2

Roll Rate Response to Pitch Rate Command: Model, Methods 1 & 2

Figure 6: Responses without command limiting (top) and with command limiting (bottom)

the knowledge of the generated moments, while others did not. More research will be needed to confirm whether relatively simple methods of command limiting may indeed be adequate for most purposes, or whether the more sophisticated methods are truly needed.

A different set of results was obtained regarding the control of systems with input and state constraints [9]. The problem under consideration was the determination of the sets of states of a linear system with control limits for which recovery to the equilibrium is possible. These results have not been applied to reconfigurable control problems yet. However, it is possible that tailless aircrafts with unstable modes in two axes and with limited control authority will have to be controlled with concerns of recoverability in mind. A procedure was developed to rapidly calculate recoverable sets numerically. The method was limited to two-dimensional, continuous-time systems with a single input, but it was found to provide significant computational savings over other methods.

3. Personnel

Faculty: Professor Marc Bodson.
Graduate Student: William Pohlchuck, completed his M.S. thesis with the support of the grant [2].
Graduate Student: John Petersen, started his Ph.D. thesis with the support of the grant.

4. Publications

[1] M. Bodson & J. Groszkiewicz, "Multivariable Adaptive Algorithms for Reconfigurable Flight Control," *IEEE Trans. on Control Systems Technology,* vol. 5, no. 2, pp. 217-229, 1997.

[2] W. Pohlchuck, *Robustness Issues in Reconfigurable Flight Control System Design,* M.S. Thesis, Department of Electrical Engineering, University of Utah, Salt Lake City, UT, June 1997.

[3] M. Bodson, "An Adaptive Algorithm with Information-Dependent Data Forgetting," *Proc. of the American Control Conference,* Seattle, WA, pp. 3485-3489, 1995.

[4] D. Ward, J. Monaco, & M. Bodson, "Development and Flight Testing of a Parameter Identification Algorithm for Reconfigurable Flight Control," submitted to the *AIAA Journal of Guidance, Control, and Dynamics.*

[5] M. Bodson, "An Adaptive Algorithm for the Tuning of Two Input Shaping Methods," to appear in *Automatica.* Preliminary version available in the *Proc. of the American Control Conference,* Alburquerque, NM, pp. 1340-1344, 1997.

[6] M. Bodson, "Experimental Comparison of Two Input Shaping Methods for the Control of Resonant Systems," *Proc. of the 13th World Congress of the International Federation of Automatic Control,* San Francisco, CA, vol. K, pp. 433-438, 1996.

[7] M. Bodson & W. Pohlchuck, "Integral Compensation in Adaptive Control," submitted to *IEEE Trans. on Automatic Control.* Preliminary version available in the *Proc. of the AIAA Guidance, Navigation, and Control Conference,* Paper AIAA 96-3772, San Diego, CA, August 1996.

[8] M. Bodson & W. Pohlchuck, "Command Limiting in Reconfigurable Flight Control," to appear in *AIAA Journal of Guidance, Control, and Dynamics.* Preliminary version available in the *Proc. of the AIAA Guidance, Navigation, and Control Conference,* New Orleans, LA, pp. 727-736, 1997.

[9] J. Stephan, M. Bodson, & J. Lehoczky, "Calculation of Recoverable Sets for Systems with Input and State Constraints," submitted to *Optimal Control Applications & Methods.*

APPENDIX

Robustness Issues in Reconfigurable Flight Control System Design

William A. Pohlchuck

A Thesis
Submitted in Partial Fulfillment
of the Requirements for
the Degree of Master of Science
in Electrical Engineering

University of Utah
Salt Lake City, Utah

June 1997

Contents

Acknowledgments

Effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-95-1-0341. The U.S. government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. government.

Abstract

The problem of reconfigurable flight control is investigated, focusing on model reference adaptive control and a stabilized recursive least-squares algorithm. Previous work has demonstrated this controller to be capable of maintaining control of an aircraft despite actuator failures. However, extensive simulations have also shown the need for further developments of the control law in order to improve transient responses and robustness to unmodeled dynamics. Modifications to the algorithm are proposed to achieve those objectives. The effectiveness of the modifications is evaluated using a detailed jet aircraft simulation model.

Chapter 1

Introduction

A need for increased reliability and for cost reduction has encouraged the development of reconfigurable flight control laws. The goal of reconfigurable flight control is to adjust the commands to the aircraft surfaces automatically following structural damage or failure in order to maintain safety and performance. Increased reliability is achieved by exploiting the computational capabilities of modern computers and the redundancy of multiple control surfaces. Costs are also reduced by alleviating the need for back-up systems.

Reconfigurable control can be applied in numerous failure situations. An example is when one of the stabilators of an airplane becomes jammed. The controls to the other unimpeded surfaces, including the other stabilator, the ailerons, and the rudder, are modified to compensate for the loss of control power in the failed actuator. Maneuverability of the aircraft is restored to acceptable levels.

The approach adopted in this thesis is based on adaptive control theory. Adaptive control provides two major benefits. First, the failure does not have to be identified among a set of predetermined possibilities. This feature is desirable due to the numerous failures that can occur. Second, adaptive controllers can continuously adapt to varying flight conditions. Some related approaches that have been proposed so far are now reviewed.

1.1 Brief Literature Review

1.1.1 Proportional-Integral Implicit Model Following

Huang and Stengel [1] implement a controller using proportional-integral implicit model following. The controller is based on a linear quadratic (LQ) control approach which utilizes a cost function that penalizes the states, the inputs, and the couplings between the states and the input. Based on the output error integral, an LQ controller equation is developed which provides both a proportional and integral feedback of error.

Huang and Stengel state the benefits of linear quadratic control theory as (1) a direct treatment of multiple-input, multiple-output systems, (2) an easy transfer from continuous to discrete-time descriptions, and (3) guaranteed stability when certain criteria are met. The application of implicit model following allows a relatively easy implementation with low gains and a low number of parameters. The simulations in the paper considered multiple failure types, but did not consider any cross-coupling effects between longitudinal and lateral dynamics. When an aircraft sustains damage, symmetry is usually lost, and undesirable couplings appear. A disadvantage of the linear quadratic method is that it requires penalty matrices. The assignment of penalties is usually ad-hoc and may need to be altered as a result of a failure.

1.1.2 Linear Regression

Chandler, Pachter, and Mears [2] utilize linear regression as an identification technique. The algorithm is a batch type algorithm able to handle time-varying parameters and parameters subject to change. A minimum variance estimate is derived using measured states of the aircraft. The authors state their objectives as (1) develop an efficient, static system identification method for adaptive/reconfigurable control, and (2) significantly improve parameter estimates by including a priori parameter relationship information from flight mechanics.

The paper does not specifically demonstrate control performance. However, the identification is shown to correctly determine the parameters after a failure in the aircraft. The identification algorithm is designed to operate continuously and to make use of prior information to simplify the time and amount of computation required. A linearized model is used for the flight dynamics, and cross-couplings are neglected. The identification technique is shown to work well and is similar to the one used in this research.

1.1.3 Feedback Linearization

Feedback linearization with combined feedback and feedforward control for large transport aircraft is investigated by Ochi and Kanai [3], [4]. Both feedback and feedforward control are used because of the varying speed of the control effectors: the feedforward control is applied to such components as the flaps and engines; feedback control is applied to the elevators, ailerons, and rudder. Feedback linearization uses a standard state-space model with the state-space matrices being linear combinations of constant parameters and known functions of the inputs to the system. The output matrix is comprised of nonlinear functions. The actual output must track the desired output. The problem is simplified by utilizing generic inputs which allow combining symmetric surfaces under the same command. The feedback linearization technique is applied to a large transport aircraft as opposed to a fighter aircraft. A nonlinear flight model is used, and the controller is found to be able to maintain flight. Cross-couplings, however, are not considered.

1.1.4 Neurocontrollers

A somewhat different approach to aircraft control is the use of neurocontrollers. This technique is proposed by Napolitano et al. [5] The fundamental concept of neurocontrollers is that they "learn" the flight conditions of the unfailed aircraft and apply the "learned" information to bring the aircraft back to equilibrium once the failure occurs. Napolitano et al. state this technique adapts well to nonlinear behavior. The learning is based on the extended back propagation algorithm (EBPA) which avoids the pitfalls of the standard back propagation algorithm (SBPA). Failure identification is necessary with this controller. Once a failure is detected, the controller switches from the "learning" mode to the "controlling" mode. Control after failure is based upon following a model. A cost function is implemented in this controller. In the unfailed case, cost is based on following the actual surface control deflections. In the failed case, cost is based on the errors between the actual and desired axis angular position, rate, and acceleration.

The use of neurocontrollers for flight control is a recent development. Control of the nonlinear system is demonstrated with a feasible on-line implementation. Cross-coupling effects are considered with this controller. One issue that should raise some concern, however, is the large transients that appear in the simulations after a failure occurs. Another issue that should raise some concern is that of "learning time." Napolitano et al. state the EBPA algorithm has a shorter learning time than the SBPA algorithm. However, any learning time could cause the aircraft to become unrecoverable if a failure occurs before the algorithm acquires sufficient information. The simulations in the paper showed a "learning time" of about 20 seconds which is quite long.

1.2 Approach of the Thesis

Model reference adaptive control is considered for solving the reconfigurable flight control problem. Benefits of the algorithm include

- the simplicity of the algorithm,
- the capability to handle cross-couplings, and
- the state feedback nature of the control law.

Flight control algorithms should be simple. The computational capability of on-board computers is limited, and calculations should be kept to a minimum. In addition, keeping an algorithm simple reduces the amount of difficulty in trouble-shooting. Cross-couplings also are significant in flight control for failed aircrafts. An unfailed aircraft is normally symmetric, resulting in a decoupling between longitudinal and lateral variables. However, in times of failure, the symmetry is usually lost. The effects of cross-couplings require a multivariable control law. Finally, the state-feedback nature of the control law is well-suited for the flight control problem because of the availability of state measurements. Stability is also better understood for state-feedback control than, for example, for a predictive control law.

1.3 Review of Previous Work

Work previous to this thesis was performed in [6]. There were three main goals for the system:

- selection of proper trim values for the inputs, even if a failure caused rapid changes in those values,
- decoupling of the inputs and outputs, and
- tracking of the reference inputs.

The realization of these three goals was achieved in [6]. However, the set of conditions for which the performance of the algorithm was evaluated was very small. Therefore, an original objective of this thesis was to expand the evaluation of the adaptive algorithms for a broader range of flight conditions and to more demanding maneuvers. In the process, unstable behavior of the closed-loop system was discovered in several instances. The problem was complicated by the fact that no single cause could be identified for the instability. The following factors were determined to be contributing:

- *low excitation:* the parameters of the control law were found to diverge during times when excitation was not provided in certain axes. The parameters were not accurately identified, resulting in inadequate control of the aircraft. If excitation was provided to all the inputs, the instabilities disappeared.

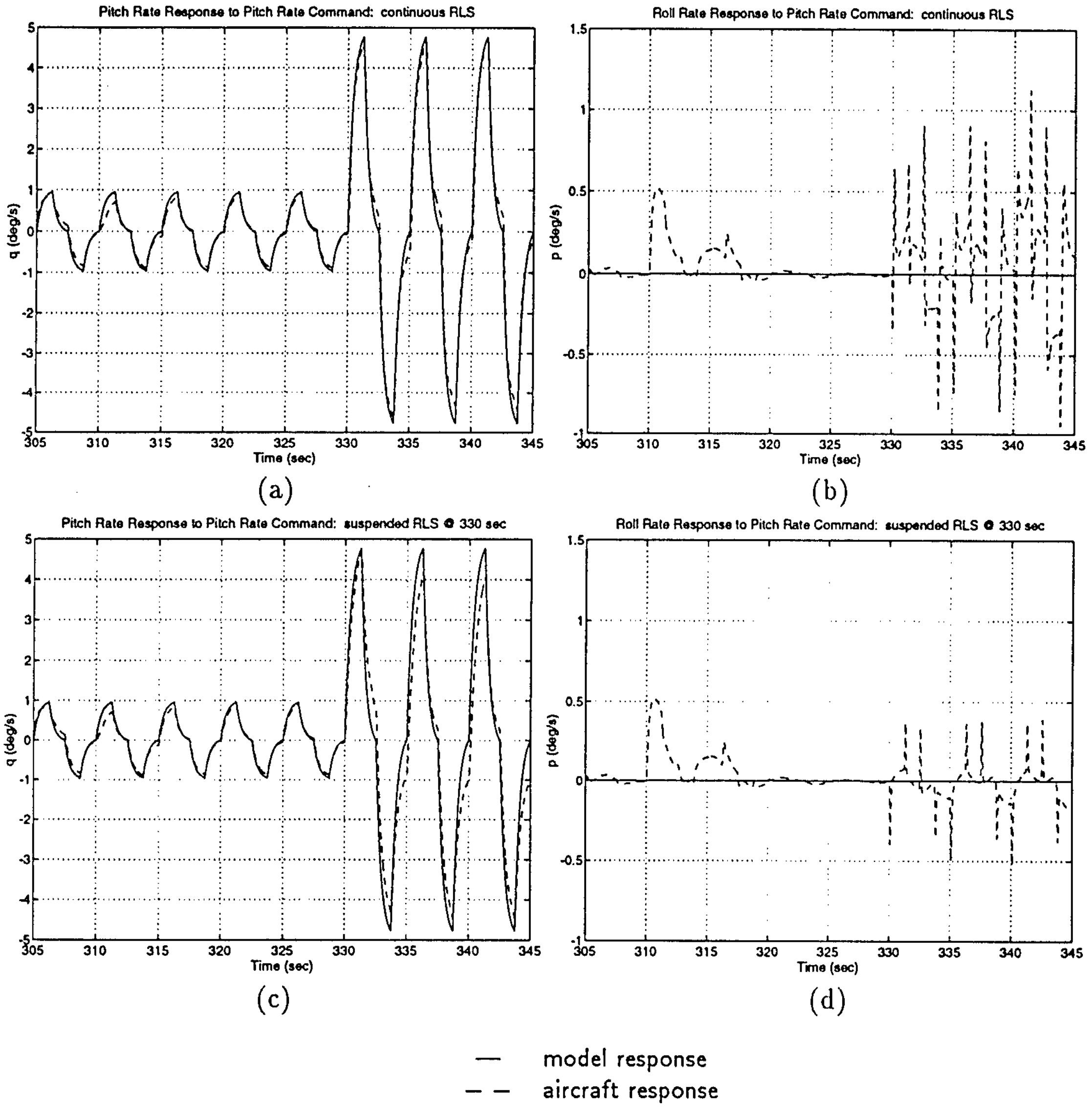
- *actuator dynamics:* the adaptive control law assumed that the control surfaces moved instantaneously. However, the actuators of the aircraft did not respond instantaneously, and the time delay of the actuators was found to contribute to the instabilities.
- *actuator limits and rate limits:* the control law calculated control inputs without constraints on the input values. However, with the actuators limited in the rate and range of motion, unstable behavior resulted.

The degradation of performance due to unmodeled nonlinearities such as actuator limits was sometimes found to be worse than for the nonadaptive algorithms. An adverse interaction between the identification and the control law developed: when the parameters generated by the identification began to diverge, control inputs actually caused the parameters to diverge more severely. Figure 1.1 illustrates this case. The left stabilator becomes locked at 310 seconds. Steps of ±1°/s are made in the pitch axis from 305 seconds to 330 seconds. This gives time for the parameters to adapt to the failure. At 330 seconds, the step size is increased to ±5°/s. In plots (a) and (b), the adaptive algorithm executes for the entire time period. In plots (c) and (d), the adaptive algorithm is suspended at 330 seconds, and the control parameters remain constant beyond 330 seconds. As can be seen in plot (b), when the adaptive algorithm is not suspended, the roll couplings become large and even appear to grow when the ±5°/s step is applied. When the adaptive algorithm is suspended, the coupling in the roll axis becomes larger than before the ±5°/s step is applied but is smaller than the continuous adaptive case and remains within a constant envelope, as shown in plot (d). Note that even though the coupling is lower in the suspended adaptive case, increasingly poor tracking results during the suspended portion in the pitch rate response in plot (c). Therefore, there is also a trade-off between tracking and coupling.

The improvements implemented in regards to instability are addressed in Chapter 3. Chapter 4 proposes four methods of command limiting to solve the actuator position and rate limiting problem. Chapter 5 addresses the issue of steady-state tracking since a steady-state tracking error has been found to occur during certain experiments. Chapter 6 considers a more realistic flight simulation environment by evaluating the performance of the aircraft with sensor noise. The emphasis of this thesis uses simulations of one flight condition and one particular failure. However, Chapter 7 illustrates the performance of the adaptive controller under different flight conditions and failures. Finally, some miscellaneous observations are made in Chapter 8, and overall conclusions are made in Chapter 9.

1.4 Summary

After considering different types of reconfigurable control, model reference adaptive control has been selected for the work of this thesis. Model reference adaptive control is simple, is able to handle cross-couplings, and is of a state feedback nature. Previous work has demonstrated that model reference adaptive control is useful for controlling an aircraft with a failure. However, robustness problems were encountered in a broad range of conditions. The focus of this thesis is to further improve the robustness of the adaptive controller.



(a) (b)

(c) (d)

Figure 1.1: Continuous Versus Suspended Adaptive Algorithm for Locked Stabilator

Chapter 2

Aircraft Model and the Basic Adaptive Control Algorithm

The purpose of this work is to investigate reconfigurable flight control. The aircraft model used in the simulations and the basic adaptive control algorithm are discussed first.

2.1 Aircraft Simulation Model

The behavior of the adaptive controller has been investigated in simulations using a model of a fighter aircraft written in FORTRAN by researchers at NASA-Dryden [7]. The simulation uses nonlinear aerodynamic data over the entire operational envelope, as well as thrust and engine response data. The aircraft model is that of a high-performance, supersonic vehicle resembling the F-15 fighter aircraft shown in Figure 2.1, except that the simulation assumes a single vertical tail surface and a single rudder.

The aircraft parameters are given in Table 2.1. The simulation model contains numerous modules: the equations of motion, the aerodynamics, the atmospheric model, the propulsion system, the aircraft actuator and surface command responses, the observation variables modeling, and a basic proportional-integral (PI) control law which may be replaced by a user-supplied control law. The relationship between the modules is shown in Figure 2.2.

The equations of motion and the aircraft aerodynamics model are six-degree-of-freedom equations representing the flight dynamics of a rigid aircraft flying in a stationary environment over a flat, nonrotating Earth. The atmospheric model calculates values for speed of sound, acceleration due to gravity, air density, viscosity, and ambient static pressure and temperature. The propulsion system is comprised of two afterburning turbofan engines.

The primary flight control surfaces are two horizontal stabilators, two ailerons, and a vertical rudder which are shown in Figure 2.3. To reduce the number of parameters that need to be identified, four commands were generated in the original code: a symmetric command to the two stabilators, a differential command to the two stabilators, a differential command to the two ailerons, and a single command to the one rudder. Table 2.2 identifies the commands for the combined surfaces, the limits imposed in the original control law, and the sign conventions. The combined commands are decomposed into individual surface commands given in Table 2.3. Also given are the position and rate limits for the commands to the surfaces. In this case, however, these limits correspond to actual physical limits and are implemented in the actuator simulation code as opposed to the control code.

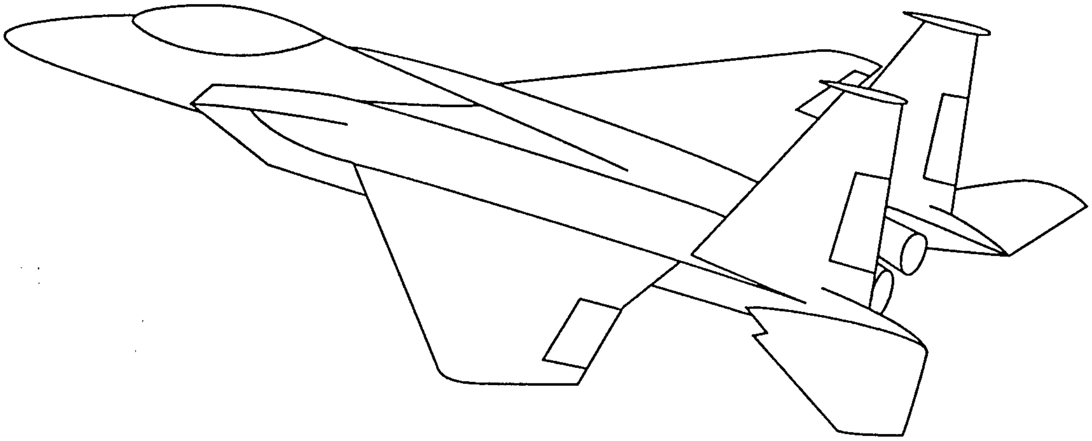


Figure 2.1: F-15 Fighter Aircraft

Table 2.1: Mass and Geometry Characteristics of Simulation Aircraft

Parameter	Symbol	Value
Wing Area	S	608.0 ft^2
Wing Span	b	42.8 ft
Mean Aerodynamic Chord	$\bar{c}$	15.95 ft
Vehicle Weight	W	45,000.0 lb
Moments of Inertia		
(Roll)	I_x	28,700.0 slug/ft^2
(Pitch)	I_y	165,100.0 slug/ft^2
(Yaw)	I_z	187,900.0 slug/ft^2
Products	I_{xz}	−520.0 slug/ft^2
	I_{xy}	0.0 slug/ft^2
	I_{yz}	0.0 slug/ft^2

Table 2.2: Command Input Definitions and Sign Conventions

Surface Command Name	Symbol	Supplied Control Law Limits (°)	Positive Sign Convention
Symmetric Stabilator	δ_{Hc}	+15/ − 25	trailing edge down
Differential Stabilator	δ_{Dc}	±20	left trailing edge down
Aileron	δ_{Ac}	±40	left trailing edge down
Rudder	δ_{Rc}	±30	trailing edge left

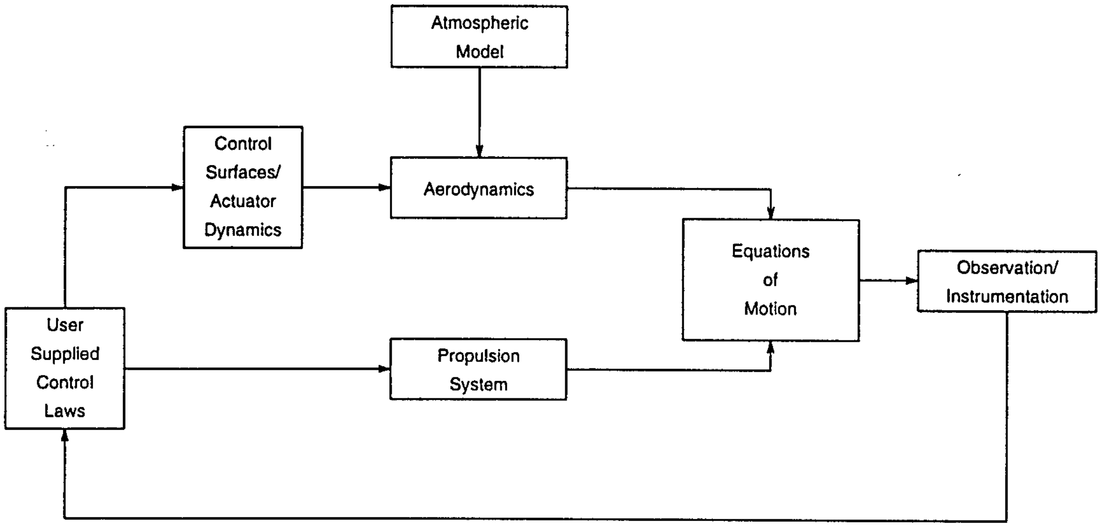

Atmospheric Model
Control Surfaces/ Actuator Dynamics
Aerodynamics
Equations of Motion
Observation/ Instrumentation
User Supplied Control Laws
Propulsion System

Figure 2.2: Modular Structure of the Model

Table 2.3: Surface Deflection Definition Equations

Surface	Deflection Definition	Position Limit (°)	Rate Limit (°/s)
$\delta_{A\text{left}_c}$	$0.5\delta_{Ac}$	±20	24
$\delta_{A\text{right}_c}$	$-0.5\delta_{Ac}$	±20	24
$\delta_{S\text{left}_c}$	$0.5(2\delta_{Hc} + \delta_{Dc})$	$+15/-25$	24
$\delta_{S\text{right}_c}$	$0.5(2\delta_{Hc} - \delta_{Dc})$	$+15/-25$	24
δ_{Rc}	δ_{Rc}	±30	24

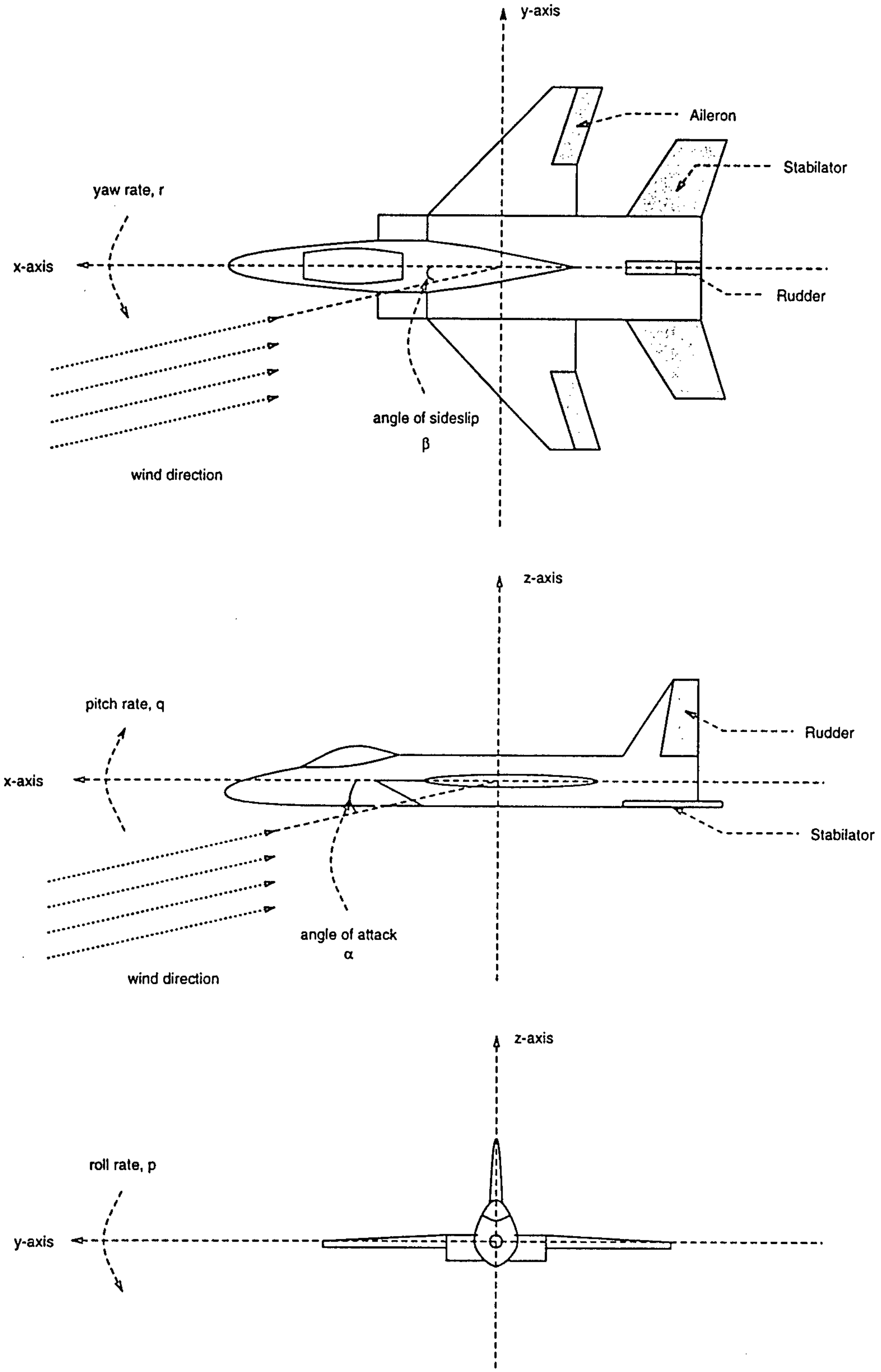


Figure 2.3: Aircraft Surfaces and State Variables

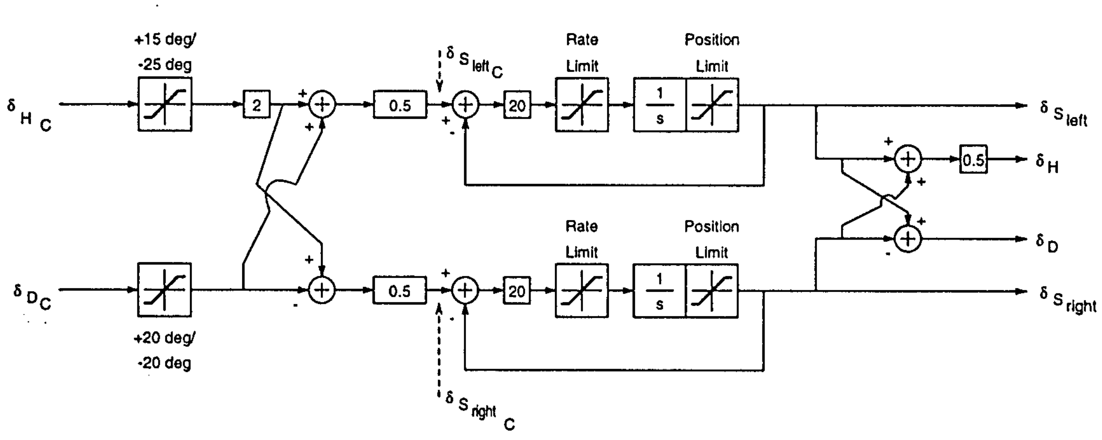


Figure 2.4: Stabilator Command Path

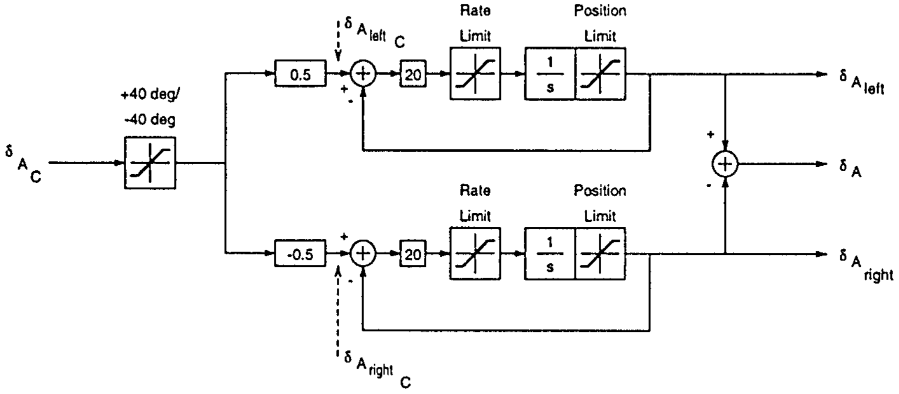


Figure 2.5: Aileron Command Path

The commands to the individual actuators are generated from δ_{Hc}, δ_{Dc}, δ_{Ac}, and δ_{Rc}. The command paths of the stabilators, ailerons, and rudder are shown in Figures 2.4 – 2.6, respectively. The feedback loops within the figures are the models for the actuators. The feedforward paths are performed by the control laws. The stabilator path is more complicated because of the symmetric and differential capability. Equivalent realized commands are also calculated based on the actual positions of each actuator. These equivalent realized commands are labeled δ_H, δ_D, δ_A, and δ_R.

The symmetric portion of the stabilator is primarily responsible for the pitching action in the aircraft. The differential portion allows the stabilator to provide additional rolling capability to that provided by the ailerons. In the original control law, the command to the differential stabilator was tied to the aileron command with the relationship as $\delta_{Dc} = \frac{5}{3}\delta_{Ac}$. With this modification, the control law and identification are simplified by only using three commands in the control law: δ_{Hc}, δ_{Ac}, and δ_{Rc}. Because of difficulties in dealing with actuator saturation with that choice, it was decided to eliminate the coupling altogether and let $\delta_{Dc} = 0$ for the simulations in this thesis unless otherwise indicated.

The observation variables are available to the control law. The supplied proportional-integral

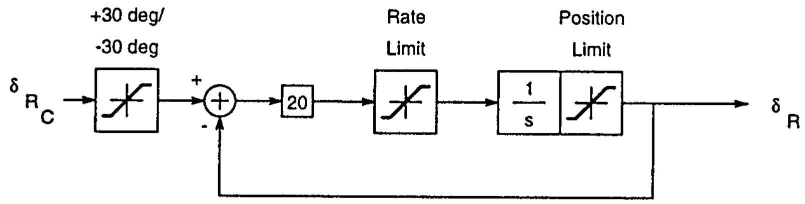


Figure 2.6: Rudder Command Path

controller can adequately fly the aircraft under ideal conditions but performs poorly in times of actuator failure. The proportional-integral controller was replaced with an adaptive control law algorithm.

2.2 System Model

For the purpose of control system design, a simplified model of the aircraft is used. It is common to consider a reduced-order model of the form

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}), \\ \mathbf{y} &= \mathbf{C}\mathbf{x}, \end{aligned} \tag{2.1}$$

where $\mathbf{x} \in R^n$, $\mathbf{u} \in R^m$, and $\mathbf{y} \in R^m$. The vector x is chosen to include five states of the aircraft ($n = 5$), with $\mathbf{x} = \begin{bmatrix} \alpha & q & \beta & p & r \end{bmatrix}^T$ indicating the angle of attack, pitch rate, angle of sideslip, roll rate, and yaw rate, respectively. The interpretation of these variables is shown in Figure 2.3. u is a three element vector ($m = 3$) of the control variables defined to be $\mathbf{u} = \begin{bmatrix} \delta_{Hc} & \delta_{Ac} & \delta_{Rc} \end{bmatrix}^T$ where δ_{Hc}, δ_{Ac}, and δ_{Rc} are the commands to the stabilators (symmetric portion), ailerons, and rudder, respectively. It is assumed that all the states are measured. The output variable $\mathbf{y} = \begin{bmatrix} q & p & r \end{bmatrix}^T$ represents the variables to be controlled.

The trim state $\mathbf{x}^*$ and the trim control input $\mathbf{u}^*$ correspond to steady, level flight and are obtained by solving

$$\begin{aligned} f(\mathbf{x}^*, \mathbf{u}^*) &= 0, \\ \mathbf{C}\mathbf{x}^* &= 0. \end{aligned} \tag{2.2}$$

A linearized model can be derived with

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}(\mathbf{x} - \mathbf{x}^*) + \mathbf{B}(\mathbf{u} - \mathbf{u}^*), \\ \mathbf{y} &= \mathbf{C}\mathbf{x}. \end{aligned} \tag{2.3}$$

With the formulation, $\mathbf{x}^*$ and $\mathbf{u}^*$ in (2.3) must be known for control design. This problem can be alleviated by representing the system with the following equations

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{d}, \\ \mathbf{y} &= \mathbf{C}\mathbf{x}, \end{aligned} \tag{2.4}$$

where

$$\mathbf{d} = \mathbf{d}^* = -\mathbf{A}\mathbf{x}^* - \mathbf{B}\mathbf{u}^*. \tag{2.5}$$

In this form, the trim uncertainty becomes a constant disturbance. Values of the parameters **A**, **B**, and **d** were determined using a least-squares identification for four flight conditions and are given in Appendix A. They were also determined in real-time using measurements of **x** and **u** and for control system design. The trim values can be determined from those parameters by solving

$$\begin{aligned} \mathbf{A}\mathbf{x}^* + \mathbf{B}\mathbf{u}^* + \mathbf{d}^* &= \mathbf{0}, \\ \mathbf{C}\mathbf{x}^* &= \mathbf{0}. \end{aligned} \tag{2.6}$$

In particular,

$$\mathbf{u}^* = -(\mathbf{CB})^{-1}\mathbf{CA}\mathbf{x}^* - (\mathbf{CB})^{-1}\mathbf{C}\mathbf{d}^*, \tag{2.7}$$

if the matrix **CB** is invertible. However, the control law does not require the explicit calculation of the trim values to be implemented.

2.3 Control Law

The specific objective of the control law considered in this thesis is to have the aircraft follow a model response specified by the system

$$\dot{\mathbf{y}}_M = \mathbf{A}_M\mathbf{y}_M + \mathbf{B}_M\mathbf{r}_M. \tag{2.8}$$

The reference input $\mathbf{r}_M$ is chosen as commanded values of angular rates, with $\mathbf{r}_M = \begin{bmatrix} q_c & p_c & r_c \end{bmatrix}^T$ and q_c, p_c, and r_c represent rate commands in pitch, roll, and yaw, respectively. The specific reference model chosen here is of the form

$$\mathbf{H}(s) = diag\{\frac{k}{s+k}\}. \tag{2.9}$$

The value of $k = 2.5$ was used in the simulations, following the suggestion of [8]. Thus, $\mathbf{A}_M$ and $\mathbf{B}_M$ in (2.8) become

$$\mathbf{A}_M = \begin{bmatrix} -2.5 & 0 & 0 \\ 0 & -2.5 & 0 \\ 0 & 0 & -2.5 \end{bmatrix}, \tag{2.10}$$

$$\mathbf{B}_M = \begin{bmatrix} 2.5 & 0 & 0 \\ 0 & 2.5 & 0 \\ 0 & 0 & 2.5 \end{bmatrix}. \tag{2.11}$$

The control objective can be achieved with the following control law

$$\mathbf{u} = \mathbf{C}_0\mathbf{r}_M + \mathbf{G}_0\mathbf{x} + \mathbf{v}_0, \tag{2.12}$$

which is shown in Figure 2.7. Indeed, substituting (2.12) in the model given by (2.4) yields

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\cdot(\mathbf{C}_0\mathbf{r}_M + \mathbf{G}_0\mathbf{x} + \mathbf{v}_0) + \mathbf{d} \\ &= (\mathbf{A} + \mathbf{B}\mathbf{G}_0)\mathbf{x} + \mathbf{B}\mathbf{C}_0\mathbf{r}_M + \mathbf{B}\mathbf{v}_0 + \mathbf{d}. \end{aligned} \tag{2.13}$$

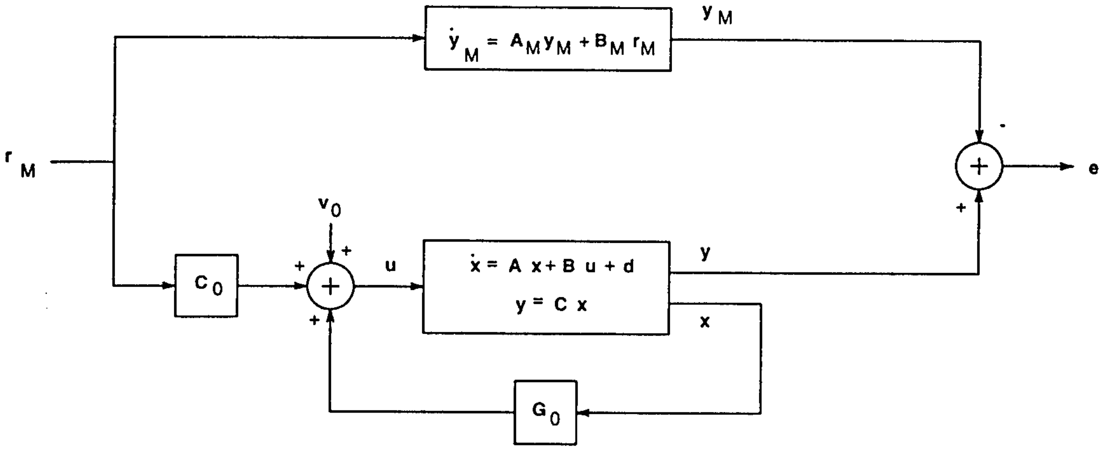


Figure 2.7: Model Reference Control Loop

Since $\dot{\mathrm{y}} = \mathrm{C}\,\dot{\mathrm{x}}$,

$$\dot{\mathrm{y}} = (\mathrm{CA} + \mathrm{CBG_0})\mathrm{x} + (\mathrm{CBC_0})\mathrm{r}_M + \mathrm{CBv_0} + \mathrm{Cd}. \tag{2.14}$$

Comparing the model $\dot{\mathrm{y}}_M = \mathrm{A}_M\mathrm{y}_M + \mathrm{B}_M\mathrm{r}_M = \mathrm{A}_M\mathrm{Cx}_M + \mathrm{B}_M\mathrm{r}_M$ with (2.14) shows that the control objective can be satisfied with the numerical matrices

$$\begin{aligned}\mathrm{C_0} &= (\mathrm{CB})^{-1}\mathrm{B}_M,\\ \mathrm{G_0} &= (\mathrm{CB})^{-1}(\mathrm{A}_M\mathrm{C} - \mathrm{CA}),\\ \mathrm{v_0} &= -(\mathrm{CB})^{-1}\mathrm{Cd},\end{aligned} \tag{2.15}$$

with $\mathrm{C_0} \in R^{m\times m}$, $\mathrm{G_0} \in R^{m\times n}$, and $\mathrm{v_0} \in R^m$. For these parameters, the response of the aircraft under the control law is identical to that of the model. The assumption is made that $(\mathrm{CB})^{-1}$ is finite [9]. The transfer function of the system (2.4) is also assumed to be minimum phase in order to guarantee internal stability of the closed-loop system.

2.4 Parameter Identification Algorithm

The thesis [6] proposed the use of an input error direct adaptive control method. An equality, proven in [6], was used as the basis for the identification technique. Specifically, one has that

$$\mathrm{u} = \mathrm{C_0^*B}_M^{-1}(\dot{\mathrm{y}} - \mathrm{A}_M\mathrm{y}) + \mathrm{G_0^*x} + \mathrm{v_0^*}. \tag{2.16}$$

This identity relates linearly unknown parameters and known signals. Therefore, standard least-squares algorithms can be used. For example, a special form of recursive least-squares algorithms with forgetting factor was studied in [10] and [11], whose error function is defined as

$$E(\theta[n]) = \sum_{k=1}^{n} \left|(\mathrm{z}[k] - \theta^T[n]\mathrm{w}[k])\right|^2 \lambda^{n-k} + \alpha\,|\theta[n] - \theta[n-1]|^2. \tag{2.17}$$

In (2.17), w and z are vectors given by

$$\mathbf{w} = \begin{bmatrix} \mathbf{B}_M^{-1}(\dot{\mathbf{y}} - \mathbf{A}_M\mathbf{y}) \\ \mathbf{x} \\ 1 \end{bmatrix}, \tag{2.18}$$

and

$$\mathbf{z} = \mathbf{u} = \begin{bmatrix} \delta_{Hc} & \delta_{Ac} & \delta_{Rc} \end{bmatrix}^T. \tag{2.19}$$

θ is a matrix of parameter estimates given by

$$\theta = \begin{bmatrix} \mathbf{C}_0^T \\ \mathbf{G}_0^T \\ \mathbf{v}_0^T \end{bmatrix}. \tag{2.20}$$

The additional term from the standard recursive least-squares algorithm is $\alpha\,|\theta[n] - \theta[n-1]|^2$. The term penalizes changes in θ. The least-squares estimate corresponding to this criterion is given by

$$\theta[n] = (\sum_{k=1}^{n} \mathbf{w}[k]\mathbf{w}^T[k]\lambda^{n-k} + \alpha\mathbf{I})^{-1}(\sum_{k=1}^{n} \mathbf{w}[k]\mathbf{z}^T[k]\lambda^{n-k} + \alpha\theta[n-1]). \tag{2.21}$$

In the identification, each column of θ is separately calculated to determine the parameters used in the calculations of each of the elements of the vector $\mathbf{z}$. Defining $\mathbf{P}[n]$ as

$$\mathbf{P}[n] = (\sum_{k=1}^{n} \mathbf{w}[k]\mathbf{w}^T[k]\lambda^{n-k} + \alpha\mathbf{I})^{-1}, \tag{2.22}$$

a recursive form for the inverse of the covariance matrix $\mathbf{P}[n]$ is

$$\mathbf{P}^{-1}[n] = \lambda\mathbf{P}^{-1}[n-1] + \mathbf{w}[n]\mathbf{w}^T[n] + \alpha(1-\lambda)\mathbf{I}, \tag{2.23}$$

with the initial condition that $\mathbf{P}^{-1}[0] = \alpha\mathbf{I}$. The recursive formula for $\theta[n]$ is

$$\theta[n] = \theta[n-1] + \mathbf{P}[n]\mathbf{w}[n](\mathbf{z}^T[n] - \mathbf{w}^T[n]\theta[n-1]) + \alpha\lambda\mathbf{P}[n](\theta[n-1] - \theta[n-2]). \tag{2.24}$$

A recursive form of $\mathbf{P}[n]$ is also desirable, so that $\mathbf{P}^{-1}[n]$ does not need to be inverted. An approximate recursive form of $\mathbf{P}[n]$ was proposed in [11] as

$$\mathbf{P}[n] = \frac{1}{\lambda}(\mathbf{P}[n-1] - \mathbf{P}[n-1]\mathbf{C}[n](\lambda\mathbf{I} + \mathbf{C}^T[n]\mathbf{P}[n-1]\mathbf{C}[n])^{-1}\mathbf{C}^T[n]\mathbf{P}[n-1]), \tag{2.25}$$

with $\mathbf{C}[n] = \begin{bmatrix} \mathbf{w}[n] & \sqrt{m\alpha(1-\lambda)}\mathbf{e}_i \end{bmatrix}$, where m is the dimension of $\mathbf{w}[n]$, and $\mathbf{e}_i$ is a vector of zeros except the i^{th} position which is one. The index i is incremented at each iteration and returns to one when the end of the vector is reached. A well-known problem in adaptive control is that precision errors in calculating $\mathbf{P}[n]$ can result in $\mathbf{P}[n]$ not necessarily being symmetric. Therefore, the symmetry of $\mathbf{P}[n]$ was used and enforced at each iteration. The stabilized recursive least-squares algorithm given by (2.24) and (2.25) was found to alleviate problems occurring during periods of low excitation, and the approximate form (2.25) was found to perform satisfactorily.

2.5 Performance Evaluation

Two types of failures were simulated in the code. The first failure is a locked surface such that the surface no longer responds to the commands and stays fixed. The second failure is a missing stabilator surface. A missing stabilator is a more severe failure than a locked stabilator because not only is control power reduced, but also the stabilization effect of the tail surface is lost. In the code, a missing stabilator was simulated by having the surface follow the direction of the local angle of attack. The local angle of attack is not exactly known, but was approximately calculated as in Appendix B. Both the locked and missing failures were separately applied to the left stabilator in the simulations. A locked aileron failure was also simulated independently, and together with a locked stabilator.

The main flight condition for the simulations is an altitude of 9,800 feet and a speed of Mach 0.5. In addition to the main flight condition, three additional flight conditions are considered in Chapter 7, as suggested by [7] to cover the operational envelope:

- 9,800 feet, Mach 0.9,
- 39,800 feet, Mach 0.6, and
- 39,800 feet, Mach 1.4.

The proportional-integral controller provided with the original code is able to maintain flight for the unfailed aircraft. In the case of the failed aircraft, the proportional-integral controller only works over limited periods of time. In both cases, however, responses of the aircraft to brief maneuvers can be used to determine the parameters of a linearized model using batch least-squares algorithms. Two different sets of the parameter matrices $\mathbf{C}_0$, $\mathbf{G}_0$, and $\mathbf{v}_0$, corresponding to the unfailed and locked stabilator conditions, were generated using a 10 second data sample. To provide excitation, steps were applied to the proportional-integral controller first in altitude, next in roll angle, and finally in the angle of sideslip. The details of the identification procedure are found in Appendix A. The results of these identification experiments are useful to evaluate the baseline controller performance independent of the adaptation response, as well as to determine nominal controller parameter values. However, the recursive algorithms do not use the off-line identification results for the failed aircraft. The state-space matrices are given for the four flight conditions in Appendix A for both the unfailed and locked stabilator aircraft with the differential stabilator movement enabled ($\delta_{Dc} = \frac{5}{3}\delta_{Ac}$). Appendix A also contains the state-space matrices for the unfailed and locked stabilator aircraft with the differential stabilator movement disabled ($\delta_{Dc} = 0$).

The performance of the control law is evaluated throughout the thesis by applying steps of commands in pitch rate or roll rate. The control law is activated at 305 seconds for both the batch and recursive identifications. The selection of 305 seconds is arbitrary, but a period of time is necessary to ensure that all transients of the simulation code have decayed. Failures occur at 305 seconds for the batch experiments and 310 seconds for the adaptive experiments. Runs are typically evaluated for 50 seconds. The aircraft code calculates the outputs of the aircraft at a rate of 50 Hz. The same sampling rate is used in the identification and the control law. For the recursive algorithms, the parameter matrices are initialized with the values from the unfailed identification case. Design parameters of the RLS controller are typically set to $\lambda = 0.99$ and $\alpha = 10.0$. Appendix C provides the details of all the simulations in the thesis.

2.6 Summary

This chapter presented the aircraft simulation model and the adaptive control algorithm that were used in the thesis, based on previous work. A reduced-order model simplified the highly-detailed, nonlinear model of a fighter aircraft. The adaptive algorithm used a model reference adaptive control law which commanded the aircraft to follow a first-order response in angular rates. The model was further simplified by designing the control law to calculate only three common control inputs. In this way, the number of parameters that needed to be identified was reduced. Parameters were identified using an indirect error equation technique and a stabilized recursive least-squares procedure. The performance of the adaptive control algorithm was demonstrated through the aircraft simulations.

Chapter 3

Modifications to Improve Robustness

The formulation of the control law is given by (2.12). The matrices C_0, G_0, and v_0 are replaced by the estimates provided by the stabilized recursive least-squares algorithm with forgetting factor. In the course of the research, certain problems developed with this implementation. New formulations to identify parameters were created and found to eliminate many of these problems. First, an inverse controller formulation was developed to address a singularity problem occurring in the C_0 matrix. In some cases, the C_0 matrix approached singularity, and instability occurred shortly thereafter. The problem was largely eliminated by identifying the inverse of the C_0 matrix, instead of C_0. Eventually, this formulation was transformed into a form of indirect identification which led to the best results overall.

Other modifications include the filtering of the actuator commands, to account for the delay in the actuator responses, and the normalization in the algorithm, to improve responses with demanding flight maneuvers. Another modification, the use of actuator measurements in the identification, was experimented with but not retained.

3.1 Indirect Identification Formulation

The adaptive algorithm is not guaranteed to be stable unless certain modifications are incorporated to ensure that the C_0 matrix is nonsingular [12]. Unfortunately, these modifications are very difficult to implement. A simple modification that was found to alleviate the problems is that of an "inverse" identification formulation. The inverse C_1 of the C_0 matrix is identified. The estimate matrix C_1 is then inverted during each iteration of the algorithm to give C_0 which is used in the control law. The original error equation is of the form

$$\mathbf{w}_0 = \mathbf{C}_0^* \mathbf{w}_1 + \mathbf{G}_0^* \mathbf{w}_2 + \mathbf{v}_0^*, \tag{3.1}$$

where

$$\begin{aligned} \mathbf{w}_0 &= \begin{bmatrix} \delta_{Hc} & \delta_{Ac} & \delta_{Rc} \end{bmatrix}^T, \\ \mathbf{w}_1 &= \mathbf{B}_M^{-1}(\dot{\mathbf{y}} - \mathbf{A}_M \mathbf{y}), \\ \mathbf{w}_2 &= \begin{bmatrix} \alpha & q & \beta & p & r \end{bmatrix}^T. \end{aligned} \tag{3.2}$$

A new equation is generated with the output based on the model and the surface commands being part of the input. Essentially, $\mathbf{w}_0$ and $\mathbf{w}_1$ are "swapping" positions in (3.1). The new equation is defined as

$$\mathbf{w}_1 = \mathbf{C}_1^* \mathbf{w}_0 + \mathbf{G}_1^* \mathbf{w}_2 + \mathbf{v}_1^*. \tag{3.3}$$

In order to obtain the controller parameters from the adaptive parameters, the relationship between the nominal parameters is used:

$$\begin{aligned} \mathbf{C}_0 &= \mathbf{C}_1^{-1}, \\ \mathbf{G}_0 &= -\mathbf{C}_1^{-1}\mathbf{G}_1, \\ \mathbf{v}_0 &= -\mathbf{C}_1^{-1}\mathbf{v}_1. \end{aligned} \tag{3.4}$$

The resulting values for $\mathbf{C}_0$, $\mathbf{G}_0$, and $\mathbf{v}_0$ are applied to the control law (2.12). Using the original formulation, the vector $\mathbf{w}[n]$ and $\mathbf{z}[n]$ are defined by (2.18) and (2.19), respectively. Using the inverse formulation, one has instead

$$\mathbf{w}[n] = \begin{bmatrix} \delta_{H_c} \\ \delta_{A_c} \\ \delta_{R_c} \\ \mathbf{x} \\ 1 \end{bmatrix}, \tag{3.5}$$

and

$$\mathbf{z}[n] = \mathbf{B}_M^{-1}(\dot{\mathbf{y}} - \mathbf{A}_M \mathbf{y}). \tag{3.6}$$

The inverse formulation is more complex than the original formulation primarily because $\mathbf{C}_1^{-1}$ must be calculated at each iteration. Further, $\mathbf{C}_1$ could be singular in this algorithm, so that $\mathbf{C}_0$ would be unbounded. However, this situation was not found to occur in simulations, so that the additional computations were worth the cost. Simulations showed that the inverse control law formulation performs better than the original control law formulation over various conditions.

The inverse controller formulation is closely related to a special form of indirect identification applicable to the model reference control problem. Typically, for an indirect identification, the state-space model

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{d} \tag{3.7}$$

is used to estimate the parameters $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{d}$. However, using (3.7) requires the identification of $(n + m + 1) \times n$ parameters, which is substantially larger than that required by the previous methods. Instead of using (3.7) in the identification,

$$\dot{\mathbf{y}} = \mathbf{C}\,\dot{\mathbf{x}} = \mathbf{C}\mathbf{A}\mathbf{x} + \mathbf{C}\mathbf{B}\mathbf{u} + \mathbf{C}\mathbf{d} \tag{3.8}$$

can be used to determine $\mathbf{CA}$, $\mathbf{CB}$, and $\mathbf{Cd}$, which are the matrices needed to calculate the controller parameters, and then only $(n + m + 1) \times m$ parameters are required. For $n = 5$ and $m = 3$, 45 parameters are required using (3.7), and 27 parameters are required using (3.8). In the latter case, the number of parameters is the same as the number of controller parameters. From the matrices $\mathbf{CA}$, $\mathbf{CB}$, and $\mathbf{Cd}$, the matrices $\mathbf{C}_0$, $\mathbf{G}_0$, and $\mathbf{v}_0$ can be calculated as in (2.15), that is

$$\begin{aligned} \mathbf{C}_0 &= (\mathbf{CB})^{-1}\mathbf{B}_M, \\ \mathbf{G}_0 &= (\mathbf{CB})^{-1}(\mathbf{A}_M\mathbf{C} - \mathbf{CA}), \\ \mathbf{v}_0 &= -(\mathbf{CB})^{-1}\mathbf{Cd}. \end{aligned} \tag{3.9}$$

For this identification formulation, the vectors required by the identification algorithm are

$$\mathrm{w}[n] = \begin{bmatrix} \delta_{Hc} \\ \delta_{Ac} \\ \delta_{Rc} \\ \mathrm{x} \\ 1 \end{bmatrix} \tag{3.10}$$

and

$$\mathrm{z}[n] = \dot{\mathrm{y}} = \begin{bmatrix} \dot{q} & \dot{p} & \dot{r} \end{bmatrix}^T. \tag{3.11}$$

Note that the identification procedure no longer depends on the model parameters $\mathbf{A}_M$ and $\mathbf{B}_M$. This fact is useful for the development of the command limiting routines presented in Chapter 4. Otherwise, the results obtained with this method were found to be identical to those obtained with the inverse controller formulation.

3.2 Filtering of Actuator Commands

Delays within the system can cause instabilities. One such location for delays is in the actuator command path. As shown in Figures 2.4 – 2.6, the equivalent realized commands (the outputs of the systems) and the surface commands (the inputs to the systems) are not identical, even if saturation is not encountered. The unity feedback system with an open loop containing an integrator and limiting routines modifies the surface command. If the position and rate limits are not encountered, the command path is modeled in the aircraft code by the first-order system

$$H(s) = \frac{20}{s+20}. \tag{3.12}$$

To increase the accuracy of the identification, the control variables δ_{Hc}, δ_{Ac}, and δ_{Rc} in (3.5) or (3.10) were filtered in the control code using (3.12). A zero-order hold approximation was used, as opposed to the Euler approximation implemented in the actuator code, so that the adaptive algorithm did not have available an exact replica of the actuator dynamics. Figure 3.1 illustrates that a low condition number of the CB matrix and stability result with the inclusion of the surface command filtering. Otherwise, instability rapidly occurs. The filtered values of the commands are only applied to the identification procedure. The unfiltered values of δ_{Hc}, δ_{Ac}, and δ_{Rc} are passed as inputs to the systems shown in Figures 2.4 – 2.6.

3.3 Use of Actuator Measurements

Another modification that contributes to the solution of the instability problem is utilizing the actuator measurements in place of the commands to the actuators. In other words, instead of using the command control variables δ_{Hc}, δ_{Ac}, and δ_{Rc} in (3.5) or (3.10), the equivalent realized commands δ_H, δ_A, and δ_R are utilized. Figure 3.2 demonstrates the effect of using actuator measurements versus using the commands to the actuators for a missing stabilator failure. Results are similar between the filtering of actuator commands and the use actuator measurements techniques. Although it would seem preferable to use the actuator measurements if they are available, the procedure has a drawback. Any change in the actuator path, such as a locking of a stabilator, will not be identified by the procedure and will have to be separately detected. With actuator filtering, the response of the actuators, surfaces, and aircraft are lumped together and the overall system dynamics are identified.

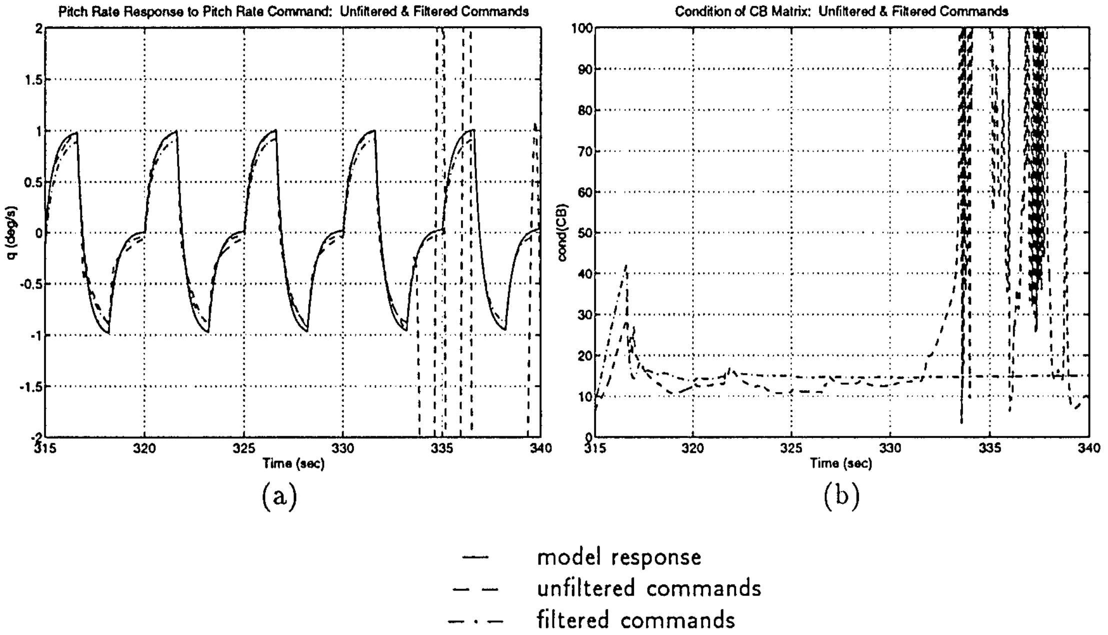


Figure 3.1: Unfiltered versus Filtered Commands for Pitch Rate Response to Pitch Rate Command

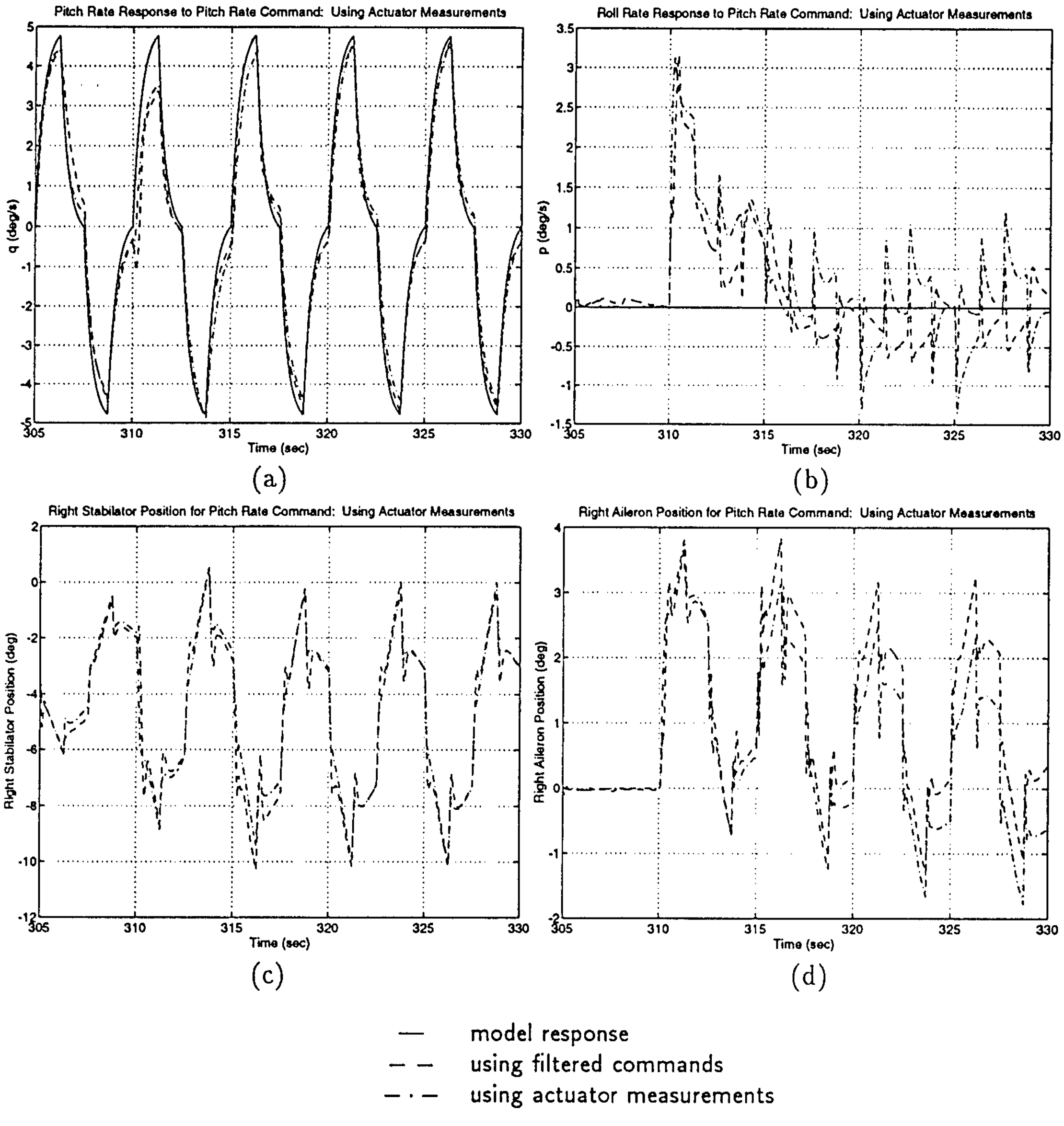
3.4 Normalization of the Algorithm

Instability worsens as the demand of the required task increases. For example, steps of ±1°/s in many cases did not generate unstable responses in the simulations. However, steps of ±5°/s or ±10°/s often resulted in instability. A contributing factor to this instability appears to be that, as the numbers in the control algorithm become large, the effect of the unmodeled dynamics is increased. A solution to this problem is to include normalization in the algorithm. The vectors $\mathbf{w}[n]$ and $\mathbf{z}[n]$ defined by (3.5) and (3.6), respectively, or (3.10) and (3.11), respectively, in the recursive least-squares algorithm are normalized by dividing each element of the vectors by $\sqrt{1+\beta\mathbf{w}^T[n]\mathbf{w}[n]}$. β is a user-defined constant. Simulations have shown that $\beta = 0.1$ yielded good results for the system.

Figure 3.3 illustrates the effect of including normalization in the algorithm for a missing stabilator failure. The glitches in the pitch rate command response in plot (a) are reduced using the normalization. The spiked responses in the roll rate coupling shown in plot (b) are also reduced. The normalization, however, is not found by itself to resolve all of the problems associated with large maneuvers and due to actuator position and rate limits. More complete solutions are discussed in the next chapter.

3.5 Summary

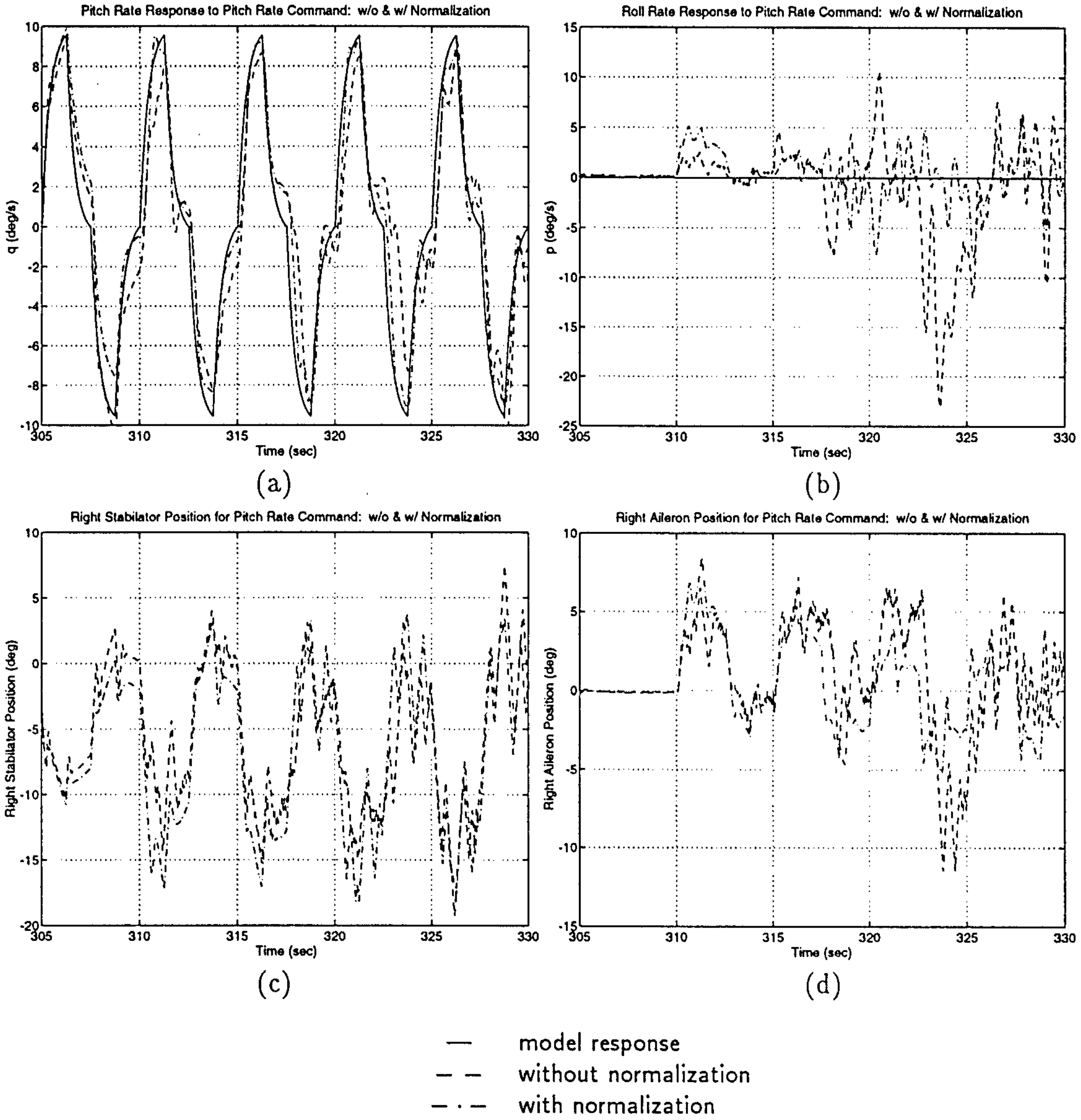
Instability problems appeared in the course of the research. Simulations showed that the method by which the parameters were identified affected the occurrence of instability. Two alternate identification formulations were proposed to reduce the likelihood of instability. The inverse controller formulation is a direct approach that uses the same control law as the original identification but requires the calculation of a matrix inverse to obtain the parameter matrices. The indirect iden-



(a) (b)

(c) (d)

Figure 3.2: Results Using Actuator Measurements for Missing Stabilator Failure



(a)

(b)

(c)

(d)

Figure 3.3: Results Using Normalization for Missing Stabilator Failure

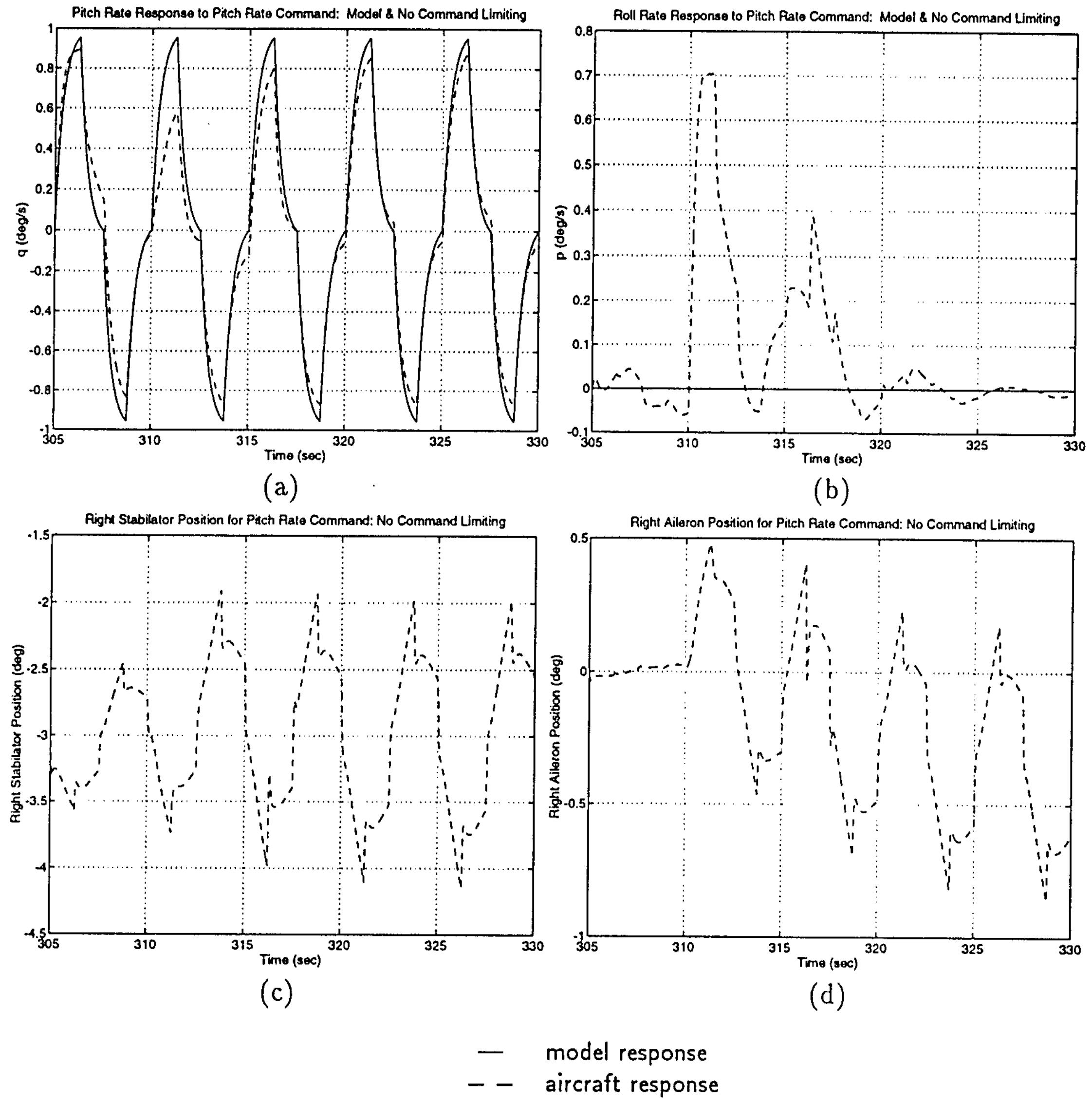
tification formulation is closely related, with the added benefit of removing the model parameters from the identification. Instability was also eliminated through the use of filtering of the actuator commands, to more accurately account for delays within the actuators, and normalization of the algorithm, to reduce the dynamic range of the signals. The use of actuator measurements improves the robustness of the system but has a drawback because it requires additional fault detection logic to handle failures related to the actuators.

Chapter 4

Actuator Position and Rate Saturation

The performance of the adaptive control law discussed so far deteriorates rapidly when the size of the steps of the reference input increases from ±1°/s to ±10°/s. Figures 4.1 and 4.2 illustrate the response of the aircraft to pitch rate commands of step sizes ±1°/s and ±10°/s, respectively. In the large step size responses, the response of the aircraft due to the pitch rate command does not match the model and has glitches that do not appear in the small step size plots. Of troubling concern is the large coupling up to ±10°/s in the roll axis, shown in Figure 4.2 (b), which does not decay with time. Part of the problem is that the actuators controlling different axes do not saturate in a coherent way, so that decoupling is lost. Therefore, the problem is not necessarily one of insufficient control power but of proper usage of the control power available. Rate saturation may be observed in the right stabilator and aileron responses. Plots (c) and (d) of Figure 4.2 show the right stabilator and aileron positions, respectively. Linear portions in the position plots are due to extensive rate limiting to ±24°/s. Plots (a) and (b) of Figure 4.3 show the corresponding rates.

There are two options to handle the saturation problem. The first is to design a control law that takes the constraints into account. For example, a constrained-optimization procedure may be applied to calculate the values of the control inputs needed to optimize some cost criterion subject to the constraints. The second is to develop a control law without considering the constraints in the design of the control law and then devising a procedure to modify the control signals when the constraints are not satisfied. For example, admissible control inputs may be calculated to approximate the desired control inputs in some optimal sense within the constraints. The first option is more difficult to develop and requires complex calculations. The second option, generally known as command limiting, has the advantages that the nominal control system design is separate from the treatment of control saturation and that the resulting algorithm is of low to moderate complexity. In general, command limiting is related to control allocation, which is the problem of distributing control requirements among multiple, constrained actuators. Control allocation has been studied in [13] – [15], whose concept is to find an allowable control input such that the generated moment in the direction of the desired moment is closest to the desired value. In [8], the idea is to find an approximation of the acceleration that is closest to the desired value in a least-squares sense. In [16], a method is proposed where the reference input is approximated so that the modified reference input applied to the control law does not yield saturation. Four control allocation methods are proposed in this thesis based on these concepts to address the saturation



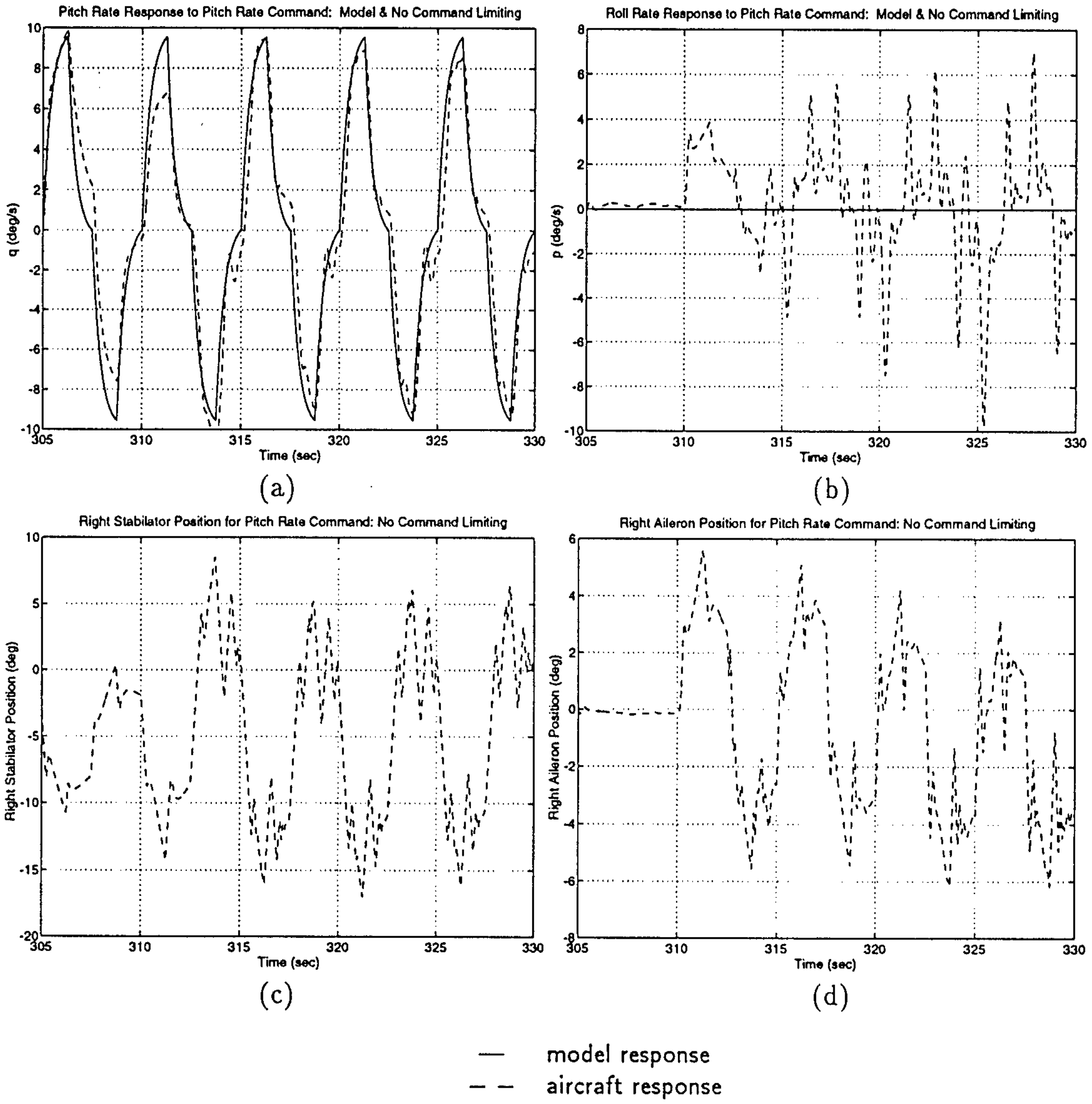
(a)

(b)

(c)

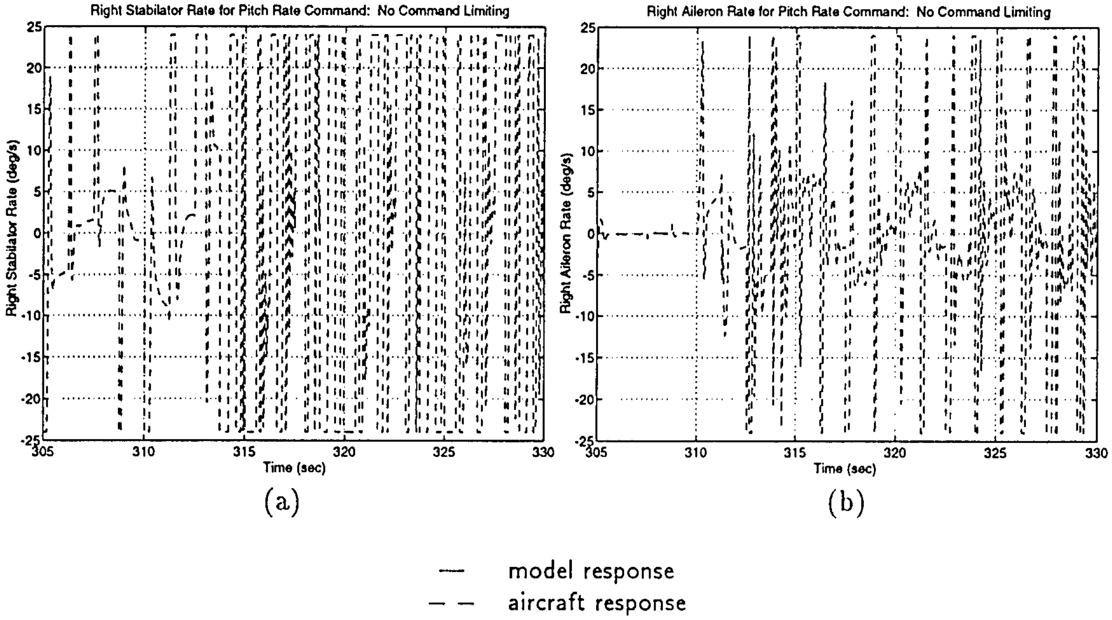
(d)

Figure 4.1: Responses to a Small Pitch Rate Command without Command Limiting



(a) (b) (c) (d)

Figure 4.2: Responses to a Large Pitch Rate Command without Command Limiting



(a)

(b)

Figure 4.3: Actuator Rates for a Large Pitch Rate Command

problem. These methods are

- scaling of control inputs,
- relaxation of control requirements,
- scaling of reference inputs, and
- a least-squares approximation to the commanded accelerations.

The methods are compared by considering computations, ease of implementation, and performance as measured by simulations. The presentation follows the organization of the paper by Bodson and Pohlchuck [17]. Although the methods were proposed earlier, their evaluation in a reconfigurable control application is novel.

4.1 Definition of Constraints

The input $\mathbf{u} = \begin{bmatrix} \delta_{Hc} & \delta_{Ac} & \delta_{Rc} \end{bmatrix}^T$ must be in the admissible position control set U_p where $U_p = \{\mathbf{u} \mid \text{ for } i = 1\ldots 3, p_{i,min} \leq u_i \leq p_{i,max}\}$. The rate of variation of $\mathbf{u}$ is also constrained so that $\mid \dot{u}_i \mid \leq d_{i,max}$ for $i = 1\ldots 3$. Because the application of digital flight control is considered, the constraint on the rate of variation can be translated into a position constraint. Given a control input $\mathbf{u}(n-1)$ at the previous time sample and a sampling period T, the admissible control rate set is defined to be U_r, where $U_r = \{\mathbf{u} \mid \text{ for } i = 1\ldots 3, u_i(n-1) - Td_{i,max} \leq u_i(n) \leq u_i(n-1) + Td_{i,max}\}$. Both the position and rate limits can be translated into a single admissible control set, $U = U_p \cap U_r$, with U of the form $U = \{\mathbf{u} \mid \text{ for } i = 1\ldots 3, u_{i,min} \leq u_i \leq u_{i,max}\}$.

4.2 Method #1: Scaling of Control Inputs

The first method is the simplest. The desired control input is denoted $\mathbf{u}_d$, and the control input produced by the command limiting technique is $\mathbf{u}$. The idea behind this method is to scale back the commands proportionally so that the constraints are satisfied. First, position constraints are checked. Of the three control elements, the one which exceeds a limit by the largest amount (if any) determines the scaling factor which will proportionally reduce the values of the other two elements. All the control elements will belong to the admissible position set after the scaling factor is applied to all three commands. Second, the rate constraints are checked, already taking into account the position limiting adjustment. All three control elements are checked to see if a rate limit is exceeded. A scaling factor is determined which reduces the largest rate excess to within acceptable limits. The same scaling factor is applied to the other two control elements so that all three elements belong to the admissible rate set. Note that by convexity of the admissible position set, it follows that the final control value still belongs to the admissible position set. The algorithm is described below.

Algorithm #1:

1. Let ρ_1 be the largest number such that $0 \leq \rho_1 \leq 1$ and $\mathbf{u}_1(n) = \rho_1 \mathbf{u}_d(n) \in U_p$.
2. Let ρ_2 be the largest number such that $0 \leq \rho_2 \leq 1$ and $\mathbf{u}_2(n) = \mathbf{u}(n-1) + \rho_2(\rho_1 \mathbf{u}_d(n) - \mathbf{u}(n-1)) \in U_r$. Let $\mathbf{u}(n) = \mathbf{u}_2(n)$.

The position limits are evaluated before the rate limits. The idea is that performing the limiting in this order is more likely to preserve the directionality of the commands.

4.3 Method #2: Relaxation of Control Requirements

The second method allows the control requirements to be relaxed when the constraints are violated. To implement this method, an alternate expression is used for the control law. The original control law is

$$\mathbf{u}_d = \mathbf{C}_0 \mathbf{r}_M + \mathbf{G}_0 \mathbf{x} + \mathbf{v}_0 \tag{4.1}$$

Substituting the values for $\mathbf{C}_0$, $\mathbf{G}_0$, and $\mathbf{v}_0$ as given by (2.15) in (4.1) yields the control law

$$\begin{aligned} \mathbf{u}_d &= (\mathbf{CB})^{-1}\mathbf{B}_M\mathbf{r}_M + (\mathbf{CB})^{-1}(\mathbf{A}_M\mathbf{C} - \mathbf{CA})\mathbf{x} - (\mathbf{CB})^{-1}\mathbf{Cd} \\ &= (\mathbf{CB})^{-1}\mathbf{B}_M\mathbf{r}_M + (\mathbf{CB})^{-1}\mathbf{A}_M\mathbf{Cx} - (\mathbf{CB})^{-1}\mathbf{CAx} - (\mathbf{CB})^{-1}\mathbf{Cd} \\ &= (\mathbf{CB})^{-1}\mathbf{B}_M\mathbf{r}_M + (\mathbf{CB})^{-1}\mathbf{A}_M\mathbf{y} - (\mathbf{CB})^{-1}\mathbf{CAx} - (\mathbf{CB})^{-1}\mathbf{Cd} \end{aligned} \tag{4.2}$$

By setting $-\mathbf{A}_M = \mathbf{B}_M = k\mathbf{I}$, (4.2) becomes

$$\begin{aligned} \mathbf{u}_d &= (\mathbf{CB})^{-1}k\mathbf{r}_M - (\mathbf{CB})^{-1}k\mathbf{y} - (\mathbf{CB})^{-1}\mathbf{CAx} - (\mathbf{CB})^{-1}\mathbf{Cd} \\ &= (\mathbf{CB})^{-1}k(\mathbf{r}_M - \mathbf{y}) + (\mathbf{CB})^{-1}(-\mathbf{CAx} - \mathbf{Cd}) \end{aligned} \tag{4.3}$$

In (4.3), $\mathbf{u}_d$ depends linearly on k. The control requirements are primarily embedded in the constant k and can be relaxed by reducing its value. Let the control vector $\mathbf{u}_d$ be decomposed as the sum of two vectors $\mathbf{u}_a$ and $\mathbf{u}_b$, given by

$$\begin{aligned} \mathbf{u}_d &= \mathbf{u}_a + \mathbf{u}_b, \\ \mathbf{u}_a &= (\mathbf{CB})^{-1}k(\mathbf{r}_M - \mathbf{y}), \\ \mathbf{u}_b &= (\mathbf{CB})^{-1}(-\mathbf{CAx} - \mathbf{Cd}). \end{aligned} \tag{4.4}$$

(4.4) gives a partition of (4.3). The idea of the method is to find a ρ that satisfies $0 \leq \rho \leq 1$ and $\rho \mathbf{u}_a + \mathbf{u}_b \in U$. In essence, ρ is reducing the value of k in the equation for $\mathbf{u}_a$ in (4.4). This method is similar to Algorithm #1, but the coding is complicated by the $\mathbf{u}_b$ term.

Algorithm #2:

1. Find the range $R_1 = [r_{1,min}, r_{1,max}]$ such that $u_{1,min} \leq \rho u_{a,1} + u_{b,1} \leq u_{1,max}$ for all $\rho \in [r_{1,min} r_{1,max}]$. The range can be obtained by considering 9 cases:

   (a) if $u_{b,1} < u_{1,min}$ and $u_{a,1} + u_{b,1} > u_{1,max}$, let

   $$\rho_{1,min} = \frac{u_{1,min} - u_{b,1}}{u_{a,1}}, \quad \rho_{1,max} = \frac{u_{1,max} - u_{b,1}}{u_{a,1}}, \tag{4.5}$$

   and $r_{1,min} = \rho_{1,min}$, $r_{1,max} = \rho_{1,max}$;

   (b) if $u_{b,1} < u_{1,min}$ and $u_{1,min} \leq u_{a,1} + u_{b,1} \leq u_{1,max}$, let $r_{1,max} = 1$ and $r_{1,min} = \rho_{1,min}$, where $\rho_{1,min}$ is defined in (4.5);

   (c) if $u_{b,1} < u_{1,min}$ and $u_{a,1} + u_{b,1} < u_{1,min}$, $R_1 = \emptyset$, the empty set;

   (d) if $u_{1,min} \leq u_{b,1} \leq u_{1,max}$ and $u_{a,1} + u_{b,1} > u_{1,max}$, let $r_{1,min} = 0$, $r_{1,max} = \rho_{1,max}$, where $\rho_{1,max}$ is defined in (4.5);

   (e) if $u_{1,min} \leq u_{b,1} \leq u_{1,max}$ and $u_{1,min} \leq u_{a,1} + u_{b,1} \leq u_{1,max}$, let $r_{1,min} = 0$, $r_{1,max} = 1$;

   (f) if $u_{1,min} \leq u_{b,1} \leq u_{1,max}$ and $u_{a,1} + u_{b,1} < u_{1,min}$, let $r_{1,min} = 0$, $r_{1,max} = \rho_{1,min}$, where $\rho_{1,min}$ is defined in (4.5);

   (g) if $u_{b,1} > u_{1,max}$ and $u_{a,1} + u_{b,1} > u_{1,max}$, $R_1 = \emptyset$, the empty set;

   (h) if $u_{b,1} > u_{1,max}$ and $u_{1,min} \leq u_{a,1} + u_{b,1} \leq u_{1,max}$, let $r_{1,max} = 1$, $r_{1,min} = \rho_{1,max}$, where $\rho_{1,max}$ is defined in (4.5);

   (i) if $u_{b,1} > u_{1,max}$ and $u_{a,1} + u_{b,1} < u_{1,min}$, let $r_{1,min} = \rho_{1,max}$, $r_{1,max} = \rho_{1,min}$, where $\rho_{1,min}$, $\rho_{1,max}$ are defined in (4.5).

2. Repeat step 1 with appropriate index changes to determine the ranges of R_2 and R_3.

3. Find $R = R_1 \cap R_2 \cap R_3$. If R is not empty, let ρ be the largest number in R and $\mathbf{u} = \rho \mathbf{u}_a + \mathbf{u}_b$. If R is empty, let $\mathbf{u}$ be such that, for $i = 1 \ldots 3$, $u_i = u_{i,min}$ if $u_{d,i} < u_{i,min}$, $u_i = u_{i,max}$ if $u_{d,i} > u_{i,max}$, and $u_i = u_{d,i}$ otherwise.

The tests are based on the values of $u_{b,1}$ and $u_{a,1} + u_{b,1}$ which are the values of $\rho u_{a,1} + u_{b,1}$ for $\rho = 0$ and $\rho = 1$, respectively. The variables $\rho_{1,min}$ and $\rho_{1,max}$ are the values of ρ such that $\rho u_{a,1} + u_{b,1}$ is equal to $u_{1,min}$ and $u_{1,max}$, respectively. These values may be outside the allowable interval, but are not calculated in that case. Division by zero in (4.5) is avoided.

A scaling factor cannot be determined in the case of R being empty in step 3. This condition is handled by applying a simple saturation function. The case occurred occasionally in simulations, but not often.

4.4 Method #3: Scaling of Reference Inputs

This method is similar to the second method, in that an input signal is scaled. Instead of scaling the control vector $\mathbf{u}_d$, the reference input $\mathbf{r}_M(n)$ is scaled. This method is computationally close to the second. Indeed, $\mathbf{u}_a$ and $\mathbf{u}_b$ were defined in (4.4). $\mathbf{u}_a$ and $\mathbf{u}_b$ can be redefined by the following equations:

$$\begin{aligned} \mathbf{u}_d &= \mathbf{u}_a + \mathbf{u}_b, \\ \mathbf{u}_a &= (\mathbf{CB})^{-1} k \mathbf{r}_M, \\ \mathbf{u}_b &= (\mathbf{CB})^{-1}(-\mathbf{CAx} - \mathbf{Cd} - k\mathbf{y}). \end{aligned} \tag{4.6}$$

The same procedure used in Method #2 is applied to this method. Again, the control vector $\mathbf{u}$ is determined such that $0 \leq \rho \leq 1$ and $\rho\mathbf{u}_a + \mathbf{u}_b \in U$. A difficulty with this technique is that it is far more common to be in a situation where no value of ρ satisfies the requirements. This problem particularly occurs when the reference input moves from a large value to a zero value. In this case, scaling does not make any difference. To resolve the problem, the algorithm can be modified so that a linear combination of the previous reference input and the current reference input is used. Instead of replacing $\mathbf{r}_M(n)$ by $\rho\mathbf{r}_M(n)$ in $\mathbf{u}_a$ given by (4.6), $\mathbf{r}_M(n)$ is replaced by the following equation:

$$\mathbf{r}_M(n) = \rho\mathbf{r}_M(n) + (1-\rho)\mathbf{r}_M(n-1). \tag{4.7}$$

For $\rho = 1$, the modified reference input is equal to the current reference input. For $\rho = 0$, the modified reference input is equal to the previous reference input. Algorithm #2 can again be applied, but with changes to the definitions of $\mathbf{u}_a$ and $\mathbf{u}_b$ in (4.6) given in the following equation:

$$\begin{aligned} \mathbf{u}_d &= \mathbf{u}_a + \mathbf{u}_b, \\ \mathbf{u}_a &= (\mathbf{CB})^{-1} k \mathbf{r}_M(n) - (\mathbf{CB})^{-1} k \mathbf{r}_M(n-1), \\ \mathbf{u}_b &= (\mathbf{CB})^{-1}(-\mathbf{CAx} - \mathbf{Cd} - k\mathbf{y}) + (\mathbf{CB})^{-1} k \mathbf{r}_M(n-1). \end{aligned} \tag{4.8}$$

The term $(\mathbf{CB})^{-1} k \mathbf{r}_M(n-1)$ is subtracted from $\mathbf{u}_a$ but is also added to $\mathbf{u}_b$. The resulting equation for $\mathbf{u}_d$ is the same as in (4.6). The algorithm is described below.

Algorithm #3:

1. Find the largest ρ such that $0 \leq \rho \leq 1$ and $\rho\mathbf{u}_a + \mathbf{u}_b \in U$, using the method of Algorithm #2, but with $\mathbf{u}_a$ and $\mathbf{u}_b$ defined in (4.6). If a ρ is found, let $\mathbf{u} = \rho\mathbf{u}_a + \mathbf{u}_b$ and stop. Else, proceed.

2. Find the largest ρ such that $0 \leq \rho \leq 1$ and $\rho\mathbf{u}_a + \mathbf{u}_b \in U$, using the method of Algorithm #2, but with $\mathbf{u}_a$ and $\mathbf{u}_b$ defined in (4.8). If a ρ is found, let $\mathbf{u} = \rho\mathbf{u}_a + \mathbf{u}_b$ and replace $\mathbf{r}_M(n)$ using (4.7). If no ρ can be found, let $\mathbf{u}$ be such that, for $i = 1 \ldots 3$, $u_i = u_{i,min}$ if $u_{d,i} < u_{i,min}$, $u_i = u_{i,max}$ if $u_{d,i} > u_{i,max}$, and $u_i = u_{d,i}$ otherwise.

Cases can still arise where a value of ρ cannot be determined. If the states and parameters vary by large amounts during a sampling period, scaling of the reference input may not bring the control input within the constraint. The simple saturation function described in Method #2 is again applied.

4.5 Method #4: LS Approximation of Commanded Accelerations

This method determines the command limiting based on the commanded accelerations. The acceleration produced is determined from the state-space model

$$\dot{\mathbf{y}} = \mathbf{CAx} + \mathbf{CBu} + \mathbf{CBd}, \tag{4.9}$$

and the component due to the desired input $\mathbf{u}_d$ is given by $\mathbf{CBu}_d$. The method, proposed by [8], consists in finding an admissible control $\mathbf{u}$ such that $\mathbf{CBu}$ optimally approximates the desired value $\mathbf{CBu}_d$. The optimization criterion is a least-squares criterion, so that the objective is to find $\mathbf{u} \in U$ such that

$$e(\mathbf{u}, \mathbf{u}_d) = \| \mathbf{CBu} - \mathbf{CBu}_d \|^2 \tag{4.10}$$

is minimized. In the absence of constraints, the solution can be obtained by setting

$$\begin{aligned} \frac{\partial}{\partial \mathbf{u}} \| \mathbf{CBu} - \mathbf{CBu}_d \|^2 &= \frac{\partial}{\partial \mathbf{u}} (\mathbf{CBu} - \mathbf{CBu}_d)^T (\mathbf{CBu} - \mathbf{CBu}_d) \\ &= 2(\mathbf{CB})^T (\mathbf{CBu} - \mathbf{CBu}_d) \\ &= 0. \end{aligned} \tag{4.11}$$

In the case of an invertible CB matrix, (4.11) yields $\mathbf{u} = \mathbf{u}_d$. With constraints, the solution can be obtained in a similar way. The algorithm considers zero, one, two, and three constraints, respectively, until a solution can be determined.

Algorithm #4:

1. If $\mathbf{u}_d \in U$, let $\mathbf{u} = \mathbf{u}_d$ and stop. Else, proceed.

2. For $u_1 = u_{1,min}$, find $\mathbf{u}$ such that $e(\mathbf{u}, \mathbf{u}_d)$ given in (4.10) is minimized without constraints on u_2 and u_3. To that effect, define $\mathbf{u}_{2,3}$ to be the vector composed of the second and third elements of $\mathbf{u}$, $(\mathbf{CB})_1$ to be the vector composed of the first column of $\mathbf{CB}$, and $(\mathbf{CB})_{2,3}$ to be the 3×2 matrix composed of the second and third columns of $\mathbf{CB}$. Then, the optimum $\mathbf{u}$ is given by

$$\mathbf{u}_{2,3} = [(\mathbf{CB})_{2,3}^T (\mathbf{CB})_{2,3}]^{-1} (\mathbf{CB})_{2,3}^T [(\mathbf{CB})\mathbf{u}_d - (\mathbf{CB})_1 u_{1,min}]. \tag{4.12}$$

   Determine the value of $e(\mathbf{u}, \mathbf{u}_d)$.

3. Repeat step 2 with the appropriate change of indices to determine the optimal $\mathbf{u}$'s constrained to $u_1 = u_{1,max}$, and similarly for $u_2 = u_{2,min}$, $u_2 = u_{2,max}$, $u_3 = u_{3,min}$, and $u_3 = u_{3,max}$. Determine the value of $e(\mathbf{u}, \mathbf{u}_d)$ in every case.

4. For $u_1 = u_{1,min}$, $u_2 = u_{2,min}$, find $\mathbf{u}$ such that $e(\mathbf{u}, \mathbf{u}_d)$ is minimized without constraints on u_3, that is

$$u_3 = [(\mathbf{CB})_3^T (\mathbf{CB})_3]^{-1} (\mathbf{CB})_3^T [(\mathbf{CB})\mathbf{u}_d - (\mathbf{CB})_1 u_{1,min} - (\mathbf{CB})_2 u_{2,min}]. \tag{4.13}$$

   Determine the value of $e(\mathbf{u}, \mathbf{u}_d)$.

5. Repeat step 4 with the appropriate change of indices to determine the optimal $\mathbf{u}$'s constrained to $u_1 = u_{1,min}$, $u_2 = u_{2,max}$ and similarly for every pair among $u_1 = u_{1,min}$, $u_1 = u_{1,max}$, $u_2 = u_{2,min}$, $u_2 = u_{2,max}$, $u_3 = u_{3,min}$, and $u_3 = u_{3,max}$. Determine the value of $e(\mathbf{u}, \mathbf{u}_d)$ in every case.

6. Determine the value of $e(\mathbf{u}, \mathbf{u}_d)$ for $u_1 = u_{1,min}$, $u_2 = u_{2,min}$, $u_3 = u_{3,min}$, and similarly for every triple among $u_1 = u_{1,min}$, $u_1 = u_{1,max}$, $u_2 = u_{2,min}$, $u_2 = u_{2,max}$, $u_3 = u_{3,min}$, and $u_3 = u_{3,max}$.

7. Collect all the candidate $\mathbf{u}$'s, eliminate those that do not belong to U, and select the one that yields the smallest $e(\mathbf{u}, \mathbf{u}_d)$.

The algorithm can be simplified if a solution is determined within the first steps. For example, step 4 is not necessary if an admissible $\mathbf{u}$ is determined in step 2. The same is true in steps 5 and 6 if a solution is obtained in the previous steps.

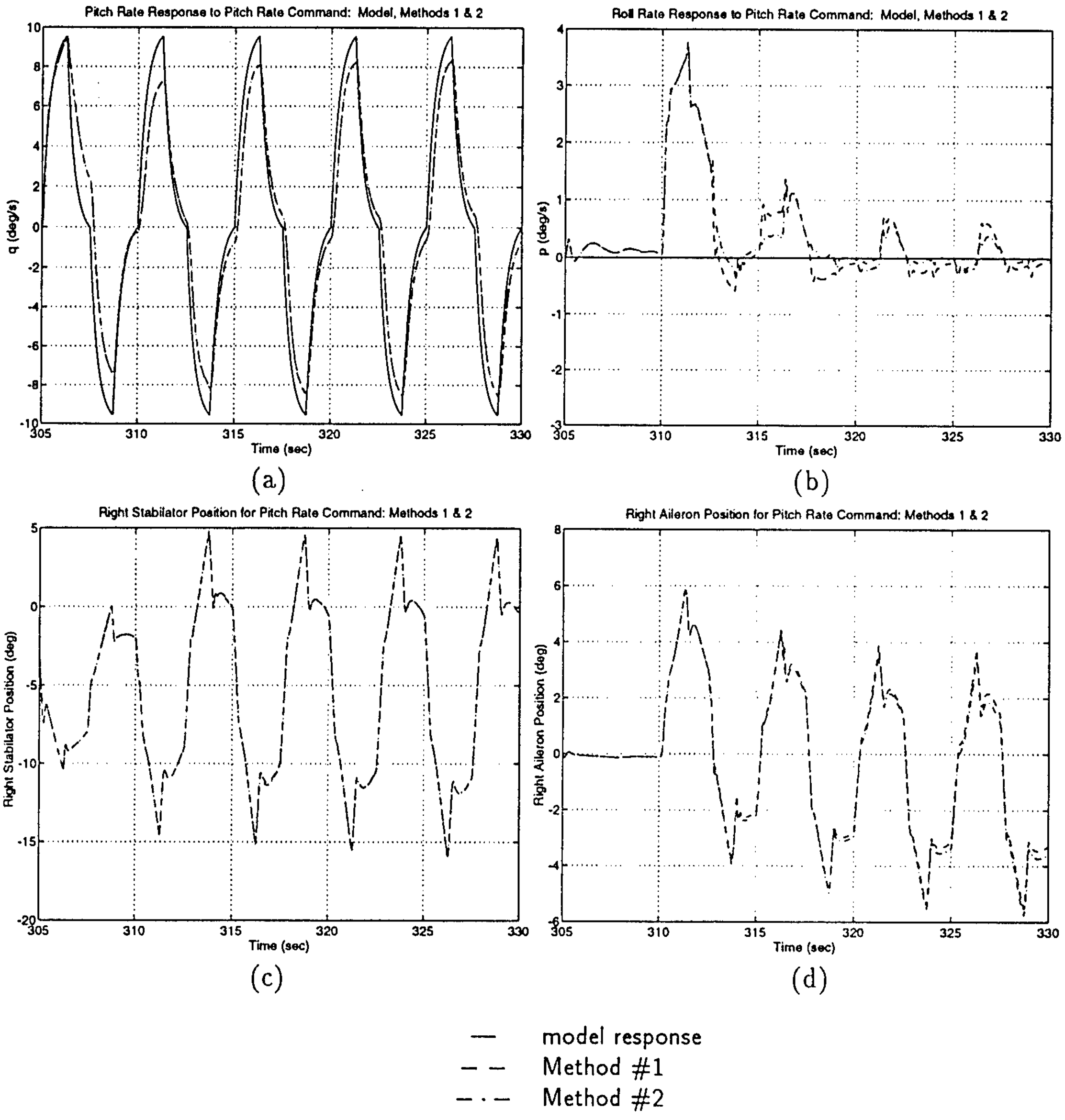
4.6 Simulation Results

Figure 4.4 shows the results of applying Methods #1 & #2. Figure 4.5 shows the results using Methods #3 & #4. Comparing the results of these figures with the results without command limiting shown in Figure 4.2, one finds that responses are significantly improved by each of the four methods. All methods perform comparably for the pitch rate responses shown in plot (a) of each figure. The ringing is eliminated, and the coupling is significantly reduced. The coupling continues to decay with time in the roll rate responses shown in plot (b) of each figure, unlike the response shown in Figure 4.2 (b). Furthermore, performance is comparable with regard to coupling for all methods, although some transient oscillations appear with Method # 4. The similar performance of the four methods is surprising since their principle and their relative complexity are so different from each other. A possible explanation is that all four methods manage to restore the directionality of the commands to a reasonable level, while simple saturation of the controls fails to do so.

The commands were limited about 20% of the time in the pitch rate command runs. For Method #2, the "no ρ" case was encountered 2% of the time, i.e., 10% of the time the command limiting routine was called. For Method #3, the "no ρ" case was encountered about 0.4% of the time.

4.7 Summary

The adaptive control algorithm had shown significantly degraded performance for larger step size maneuvers, partially due to saturation occurring in the actuators. The control law would calculate command values that were not attainable by the actuators. The problem was alleviated by developing four different methods of command limiting. Simulation results do not show one particular method to be superior to the others, although the complexity is quite different for the four methods. All four methods do, however, offer significant improvement to the performance of the aircraft. To simplify future analyses, the least-squares approximation to commanded accelerations will be used.



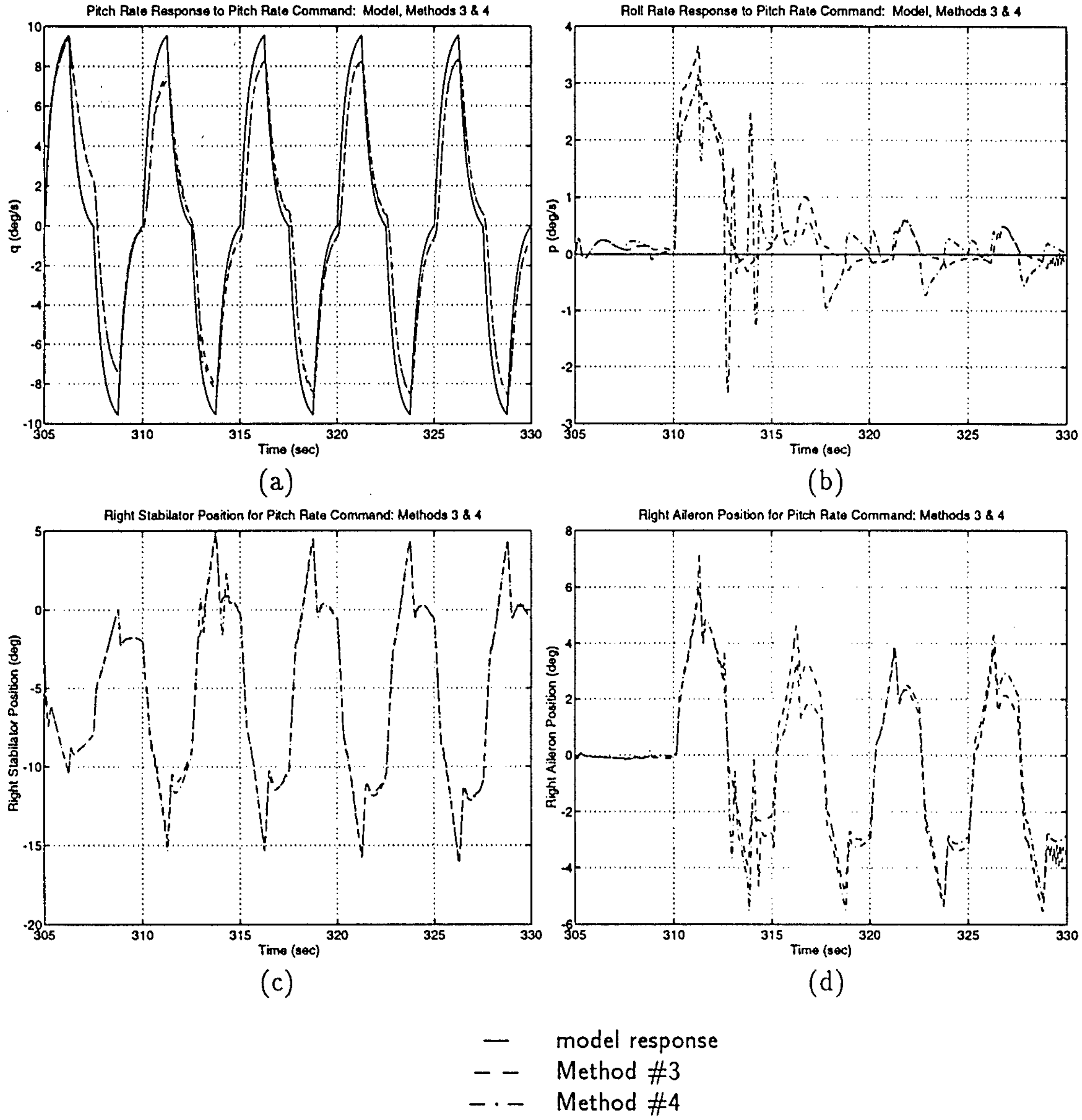
(a)

(b)

(c)

(d)

Figure 4.4: Responses to a Large Pitch Rate Command Using Methods #1 & #2



(a) (b) (c) (d)

Figure 4.5: Responses to a Large Pitch Rate Command Using Methods #3 & #4

Chapter 5

Steady-State Tracking

Integral compensation is used in nonadaptive linear control systems to ensure the tracking of constant reference inputs and the rejection of constant disturbances. Since the responses of the aircraft using the adaptive flight controller have shown small errors in steady-state tracking, particularly in times of failure, the application of integral compensation to the control law was considered. Integral compensation can be incorporated in the adaptive control algorithms discussed so far. The improvements that can be obtained with this addition will be shown.

5.1 Benefits of Integral Control

Integral compensation is reviewed in general terms in this section. A single-input, single-output plant $P(s)$ is considered with input u, output y, and input disturbance d, so that

$$y = P(s)[u + d]. \tag{5.1}$$

The term d represents a constant disturbance, i.e., $d(t) - d^*$. It was shown in Section 2.2 that the trim uncertainty could be viewed as a constant disturbance, and a major objective of the reconfigurable control system is to compensate for trim variations. A standard compensator, based on the tracking error $r_M - y$, is given by

$$u = C(s)[r_M - y], \tag{5.2}$$

where r_M is the reference input. The resulting output is

$$y = H_1(s)[r_M] + H_2(s)[d], \tag{5.3}$$

where

$$H_1(s) = \frac{P(s)C(s)}{1 + P(s)C(s)}, \qquad H_2(s) = \frac{P(s)}{1 + P(s)C(s)}. \tag{5.4}$$

Integral compensation consists in having a pole of C(s) at $s = 0$. In that case, $H_1(0) = 1$, which means that constant reference inputs are perfectly tracked, and $H_2(0) = 0$, which means that constant disturbances are perfectly rejected. Attractive features of integral compensation are that both of the objectives are achieved together, and that they are achieved despite uncertainties in the plant parameters, as long as the closed-loop system remains stable.

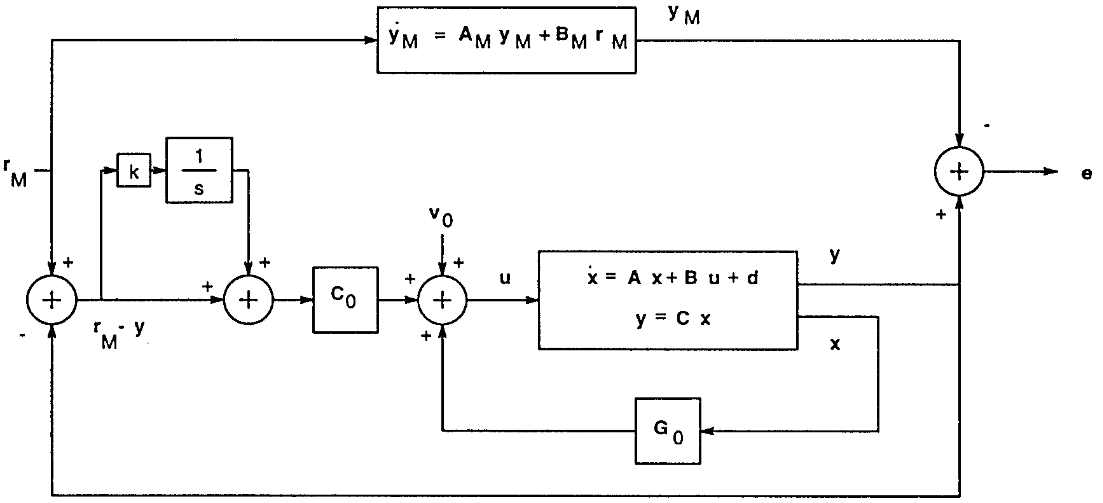


Figure 5.1: Model Reference Control Loop with Integral Compensation

5.2 Integral Control Law

In [18], Bodson and Pohlchuck proposed to add integral compensation to the control law. The original control law of (2.12) is shown in Figure 2.7. The modified control law is

$$\mathrm{u} = \mathrm{C}_0(\mathrm{r}_M - \mathrm{y} + k\int_0^t (\mathrm{r}_M - \mathrm{y})dt) + \mathrm{G}_0\mathrm{x} + \mathrm{v}_0 \tag{5.5}$$

and is shown in Figure 5.1. Assuming zero initial conditions and the aircraft model $\dot{\mathrm{y}} = \mathrm{CAx} + \mathrm{CBu} + \mathrm{Cd}$, one has, in the Laplace domain

$$\begin{aligned}(s^2 + 2ks + k^2)\mathrm{y}(s) &= \mathrm{CA}s\mathrm{x}(s) + \mathrm{CB}s\mathrm{u}(s) + \mathrm{C}s\mathrm{d}(s) + 2ks\mathrm{y}(s) + k^2\mathrm{y}(s) \\ &= (\mathrm{CA} + \mathrm{CBG}_0)s\mathrm{x}(s) + \mathrm{CBC}_0 s\mathrm{r}_M(s) - \mathrm{CBC}_0 s\mathrm{y}(s) \\ &\quad + k\mathrm{CBC}_0\mathrm{r}_M(s) - k\mathrm{CBC}_0\mathrm{y}(s) + \mathrm{CB}s\mathrm{v}_0(s) + \mathrm{C}s\mathrm{d}(s) \\ &\quad + 2ks\mathrm{y}(s) + k^2\mathrm{y}(s).\end{aligned} \tag{5.6}$$

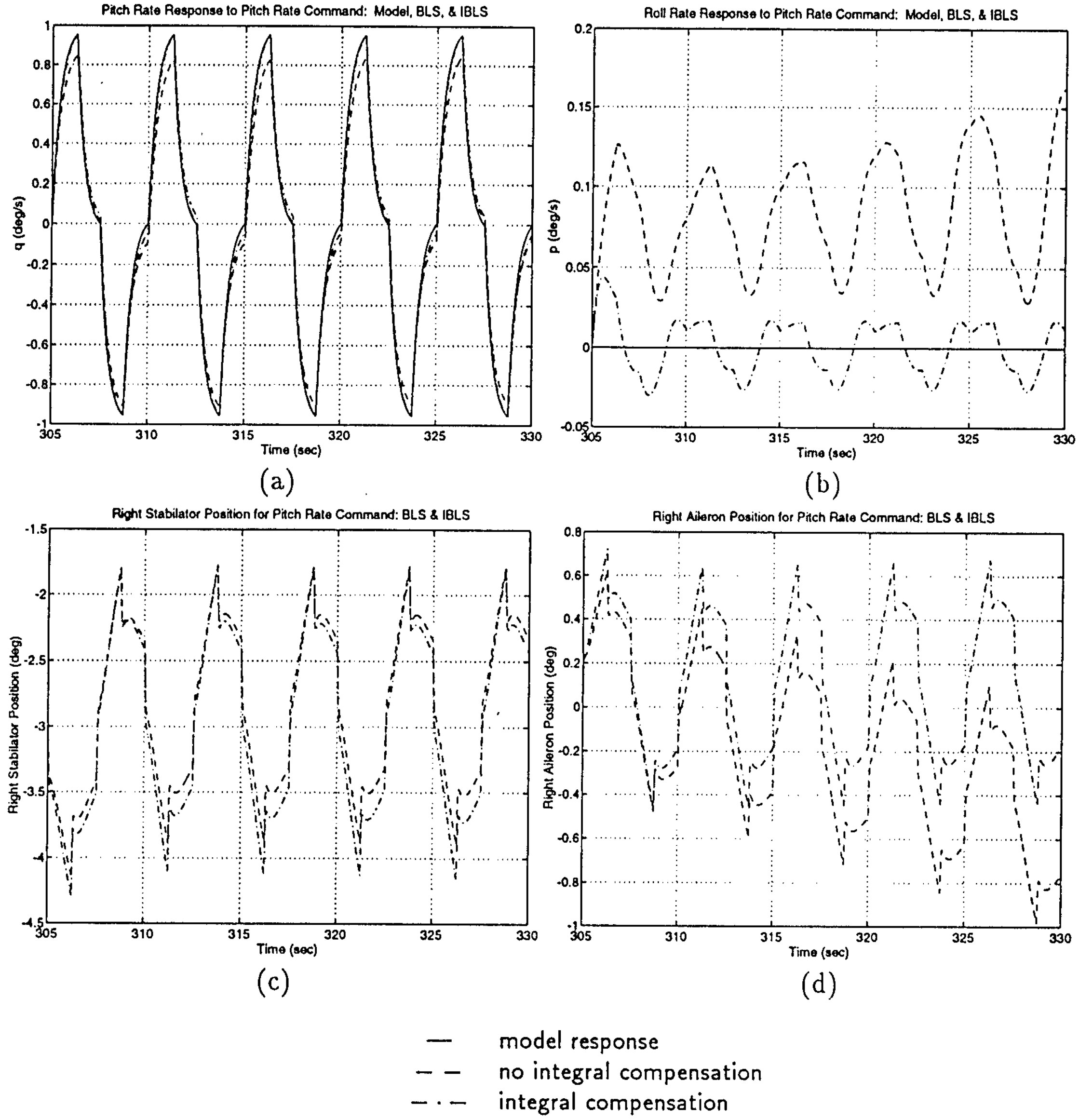
Using the nominal values of the parameters yields

$$\mathrm{y}(s) = \frac{k(s+k)}{(s^2 + 2ks + k^2)}\mathrm{r}_M(s) = \frac{k}{s+k}\mathrm{r}_M(s). \tag{5.7}$$

In other words, using the same parameter matrices, the closed-loop response remains the first-order reference model transfer function of (2.9).

5.3 Simulation Results

Figures 5.2 – 5.5 compare results with and without integral compensation. The least-squares command limiting method is applied. Responses with a stabilator locked at 305 seconds during a batch least-squares run appear in Figure 5.2. The addition of integral compensation improves


Pitch Rate Response to Pitch Rate Command: Model, BLS, & IBLS

(a)

Roll Rate Response to Pitch Rate Command: Model, BLS, & IBLS

(b)

Right Stabilator Position for Pitch Rate Command: BLS & IBLS

(c)

Right Aileron Position for Pitch Rate Command: BLS & IBLS

(d)

Figure 5.2: BLS Results for the Locked Stabilator Aircraft with and without Integral Compensation

steady-state tracking and prevents drift in this case. Interestingly, improvements in cross-coupling effects are also observed in plot (b). Thus, for the batch least-squares case, integral compensation generates near-model responses in both the pitch and roll rate.

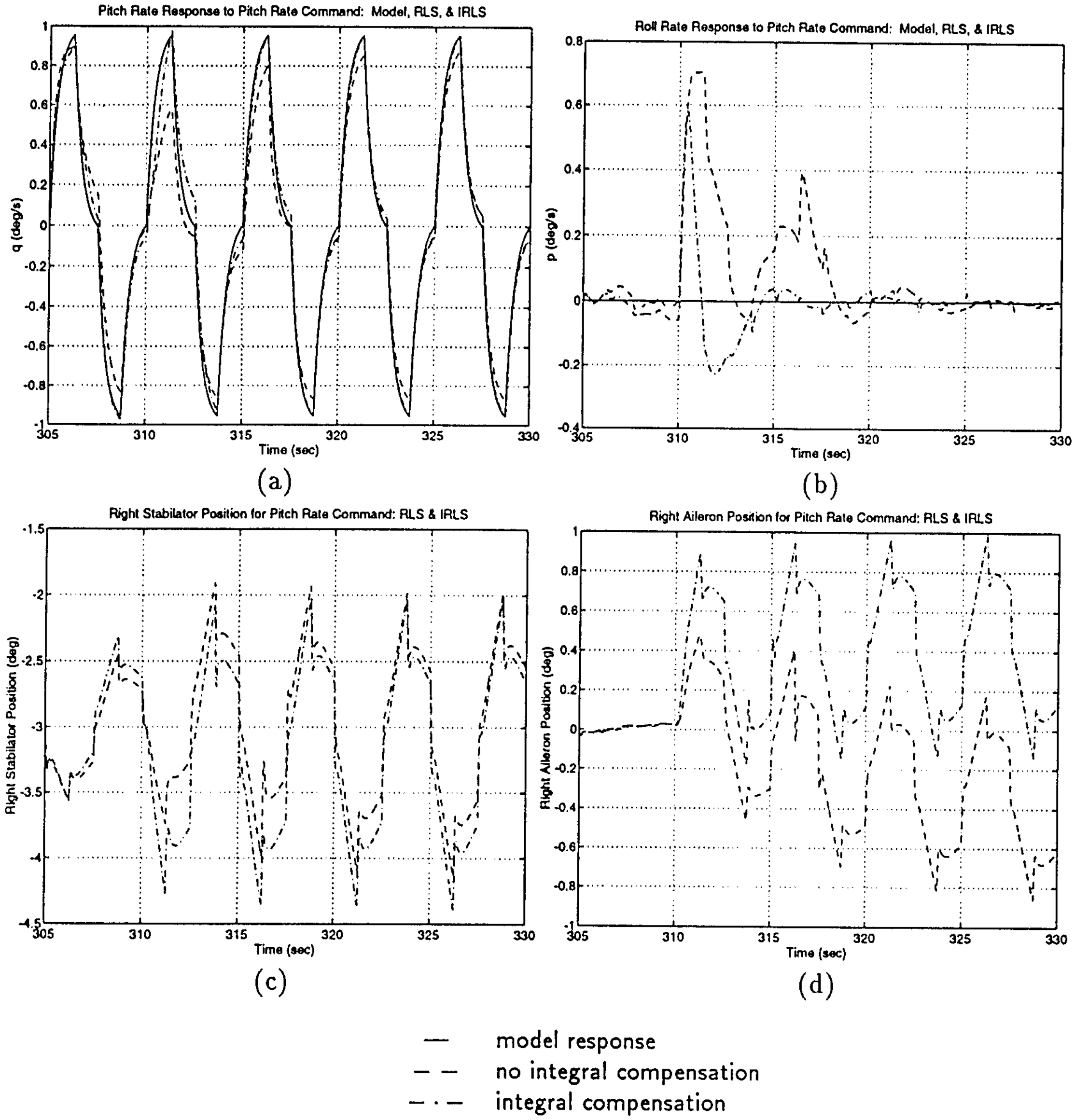
Results are shown in Figure 5.3 for the stabilized recursive least-squares control algorithm applied to an aircraft in which the left stabilator becomes locked at 310 seconds. Note that the tracking errors are slightly smaller with the recursive least-squares algorithm than with the batch least-squares algorithm. Furthermore, integral compensation causes the response of the aircraft to more quickly match the model response once a failure occurs. Observing the right stabilator and aileron positions in plots (c) and (d), respectively, shows that after the failure occurs, larger commands to both actuators result. The trim to both actuators also changes, and integral compensation prevents divergence of the aileron position.

Figure 5.4 considers the recursive least-squares control algorithm applied to the aircraft with a missing left stabilator occurring at 310 seconds. Again, integral compensation causes the model response to be matched more quickly. Similar to the previous case, the cross-couplings in plot (b) are lower with integral compensation. Note, however, a period of rapid oscillation occurs with integral compensation.

Figure 5.5 considers the locked stabilator failure for a step size of ±10°/s. Integral compensation does not adversely affect the aircraft during this maneuver. However, it does not seem to offer great improvement either. In fact, integral compensation generates a higher transient cross-coupling in plot (b). Overall, integral compensation appears to significantly improve batch least-squares responses, but only offers slight improvement in the recursive least-squares cases.

5.4 Summary

Improvements to steady-state tracking can be achieved by adding integral compensation to the control law. The most significant improvement results using the batch least-squares identification algorithm. This is compatible with conventional knowledge regarding the use of integral compensation in nonadaptive systems. Integral compensation does not offer much improvement when using the recursive least-squares identification algorithm, i.e., in the adaptive context under consideration for this application. The recursive algorithm is able to fine-tune the controller matrices and, as a result, provide some of the benefits normally attributed to integral compensation. Additionally, integral compensation using the recursive least-squares algorithm often causes the parameters to change more abruptly, thus resulting in spiked or oscillatory behavior in the aircraft output. Overall, the benefits of integral compensation are limited and probably not worth the additional complexity.



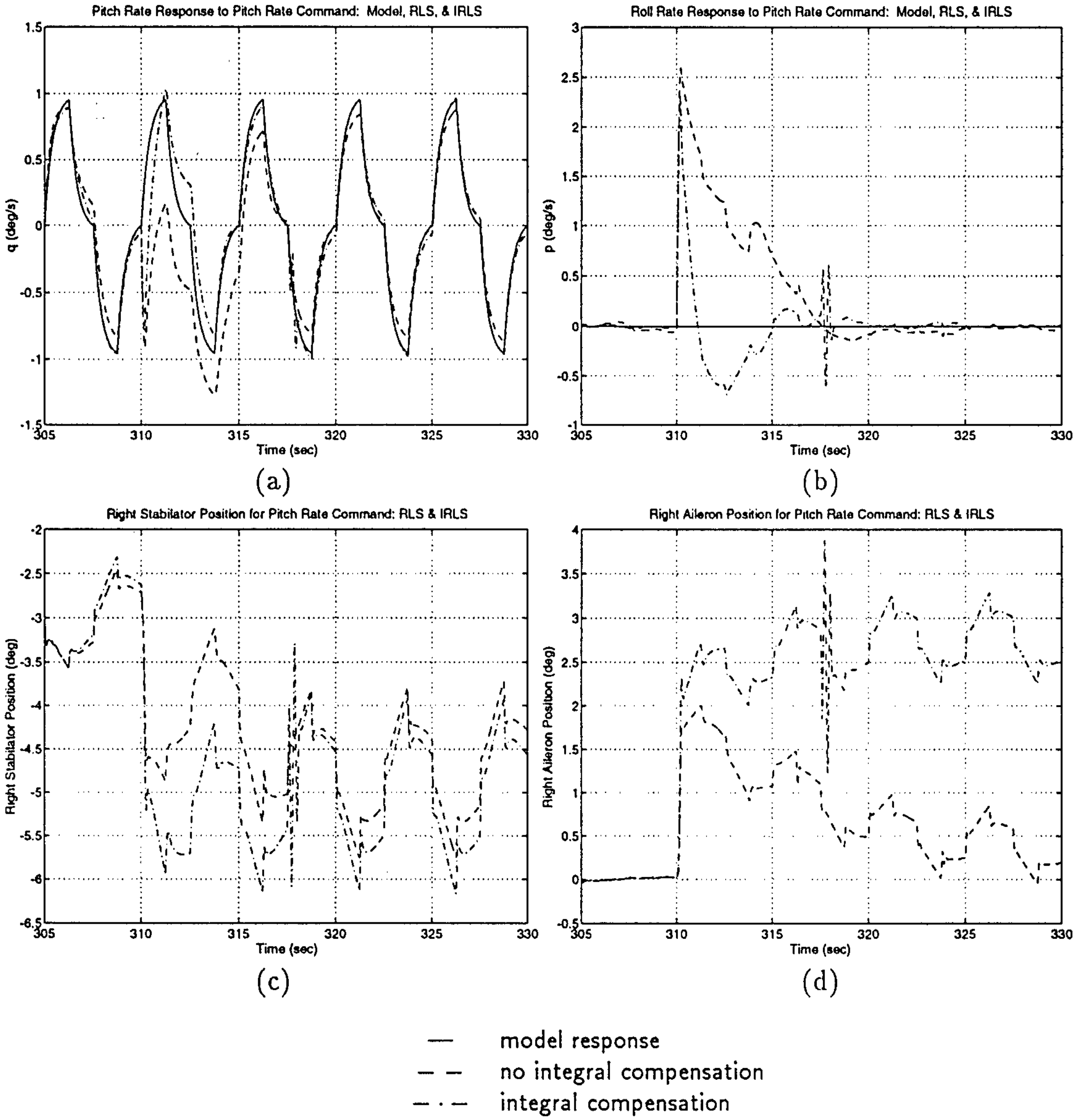
(a)

(b)

(c)

(d)

Figure 5.3: RLS Results for the Unfailed then Locked Stabilator Aircraft with and without Integral Compensation



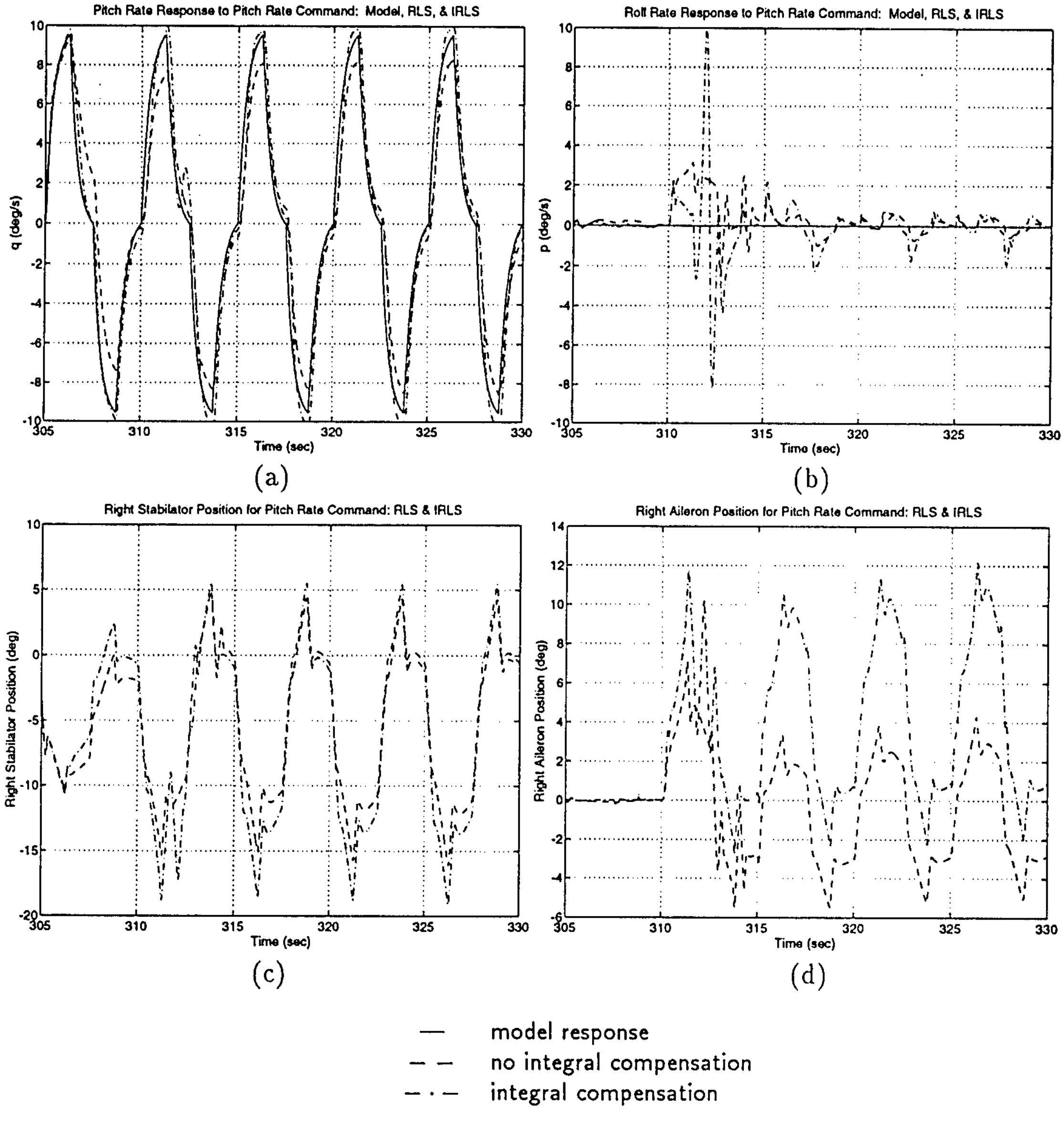
(a)

(b)

(c)

(d)

Figure 5.4: RLS Results for the Unfailed then Missing Stabilator Aircraft with and without Integral Compensation



(a)

(b)

(c)

(d)

Figure 5.5: RLS Results for the Unfailed then Locked Stabilator Aircraft for a Large Command

Chapter 6

Noise Analysis

6.1 Noise Models

Measurement devices generate signals that contain noise. The addition of noise can complicate the control problem, especially if adaptation is required. Previous work did not include the effect of noise because noise models were not part of the original simulation code. However, concern about the possible impact of the noise in a practical environment needed to be addressed. For this reason, a few experiments were performed with noise. Noise was modeled as Gaussian white noise with standard deviations determined from the literature [19] as well as flight test data, and was added to the measurements used by the adaptive algorithms and the control law. Table 6.1 lists the standard deviations used in the simulations for various signals. In the code, a Gaussian noise source is created from a uniform random number generator [20], [21] converted to a Gaussian distribution [22]. The specific routines for generating the Gaussian random numbers are described in Appendix D.

6.2 Simulation Results

Figures 6.1 and 6.2 illustrate the performance of the adaptive algorithm with and without the noise models applied for step sizes of ±1°/s and ±10°/s, respectively. The algorithm uses the least-

Table 6.1: Noise Standard Deviations

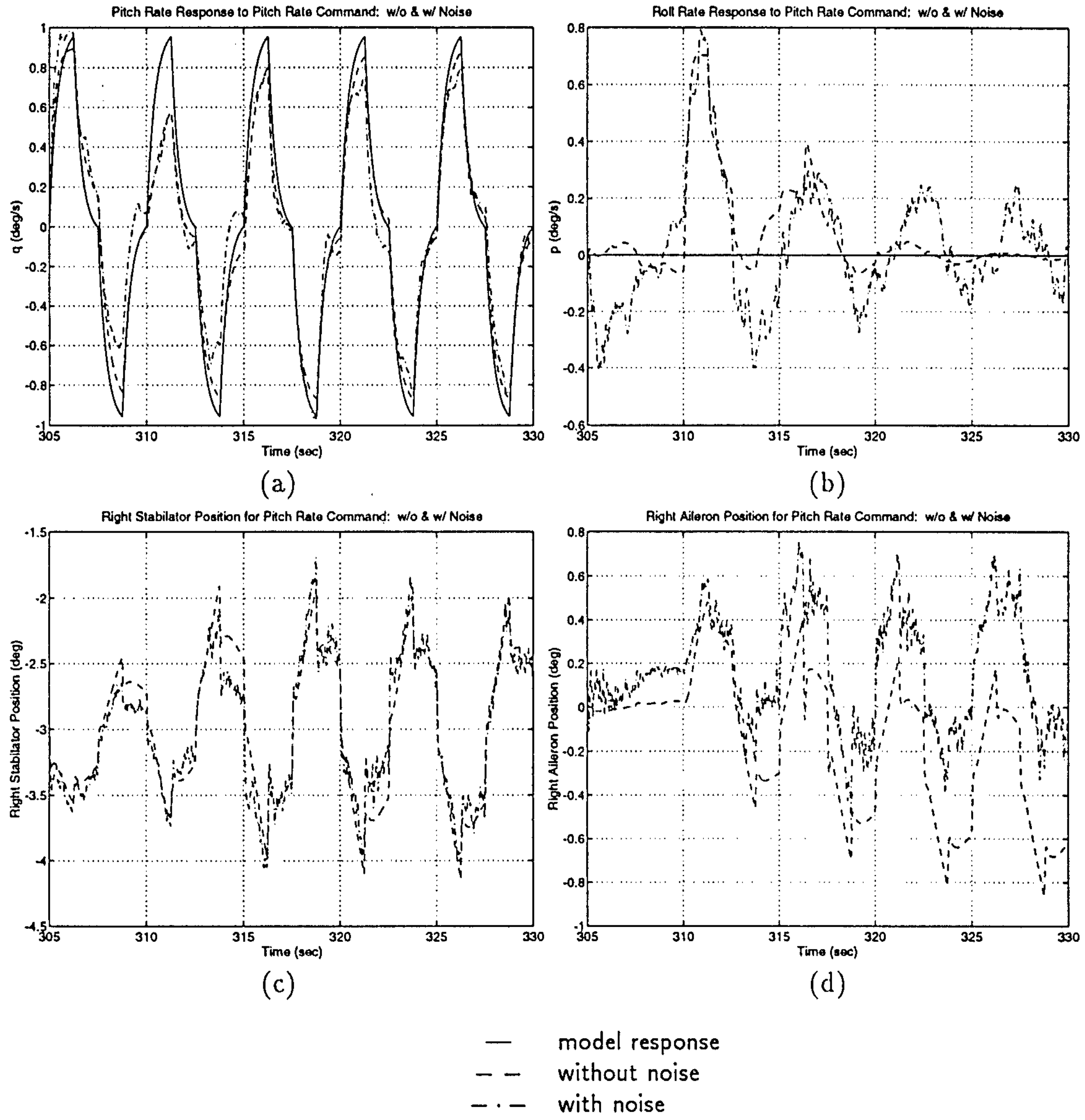
Parameter	Noise Standard Deviation
α	0.1°
β	0.1°
q	0.1°/s
p	0.1°/s
r	0.1°/s
$\dot{q}$	$1.5°/\text{s}^2$
$\dot{p}$	$3.0°/\text{s}^2$
$\dot{r}$	$1.5°/\text{s}^2$

squares approximation of commanded accelerations as the method of command limiting. Results illustrate that the algorithm handles the noise well. Results appear to be worse in the ±1°/s case because the values of the noise are similar in magnitude to the step sizes commanded. The noise does not appear to have much of an effect on the ±10°/s run. These runs are representative of the behavior of the adaptive control law observed for a number of simulations in which noise was added. No simulations with noise generated unstable behavior in the aircraft. Overall, the algorithm was found to be capable of handling the noise well.

As previously mentioned, the method of command limiting used in Figures 6.1 and 6.2 was the least-squares approximation of commanded accelerations. Simulations were also run with the other command limiting methods. Interestingly, the noise affected the amount of command limiting. For example, in Figures 4.4 and 4.5 commands were limited about 20% of the time. When noise was added to the same simulation, the commands were limited around 25% of the time. In Figures 4.4 and 4.5 the "no ρ" case was encountered about 2% of the time. With the addition of noise, "no ρ" cases occurred about 5% of the time. However, the increase in command limiting did not significantly reduce the performance of the adaptive controller.

6.3 Summary

The brief analysis of this chapter showed that noise did not affect the adaptive algorithm significantly. A simple Gaussian white model of noise added to the sensors was used. In practice, the perturbation observed in the measured signals consists primarily of quantization and unmodeled effects such as flexible modes. This noise would be more complicated to model and would be the focus of future research. The above results, however, demonstrate that the algorithm is not unduly sensitive to realistic amounts of measurement noise.



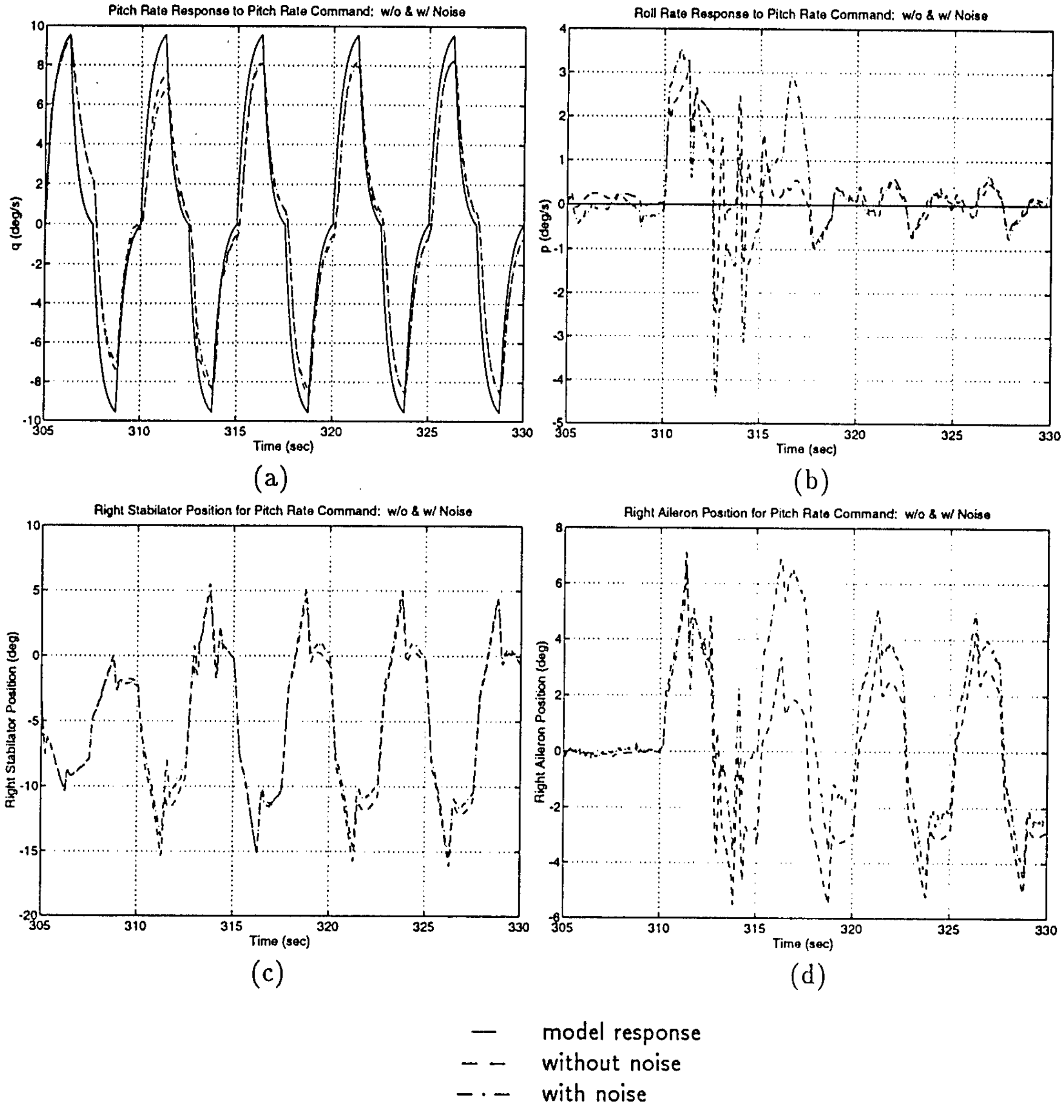
(a)

(b)

(c)

(d)

Figure 6.1: Results with and without Noise for ±1°/s Locked Stabilator Failure



(a)

(b)

(c)

(d)

Figure 6.2: Results with and without Noise for ±10°/s Locked Stabilator Failure

Chapter 7

Evaluation for Different Flight Conditions and Failures

This chapter examines the performance of the adaptive controller for other flight conditions, including two other subsonic and one supersonic condition. Additional failures, a locked aileron and a combined locked stabilator and aileron, are also examined.

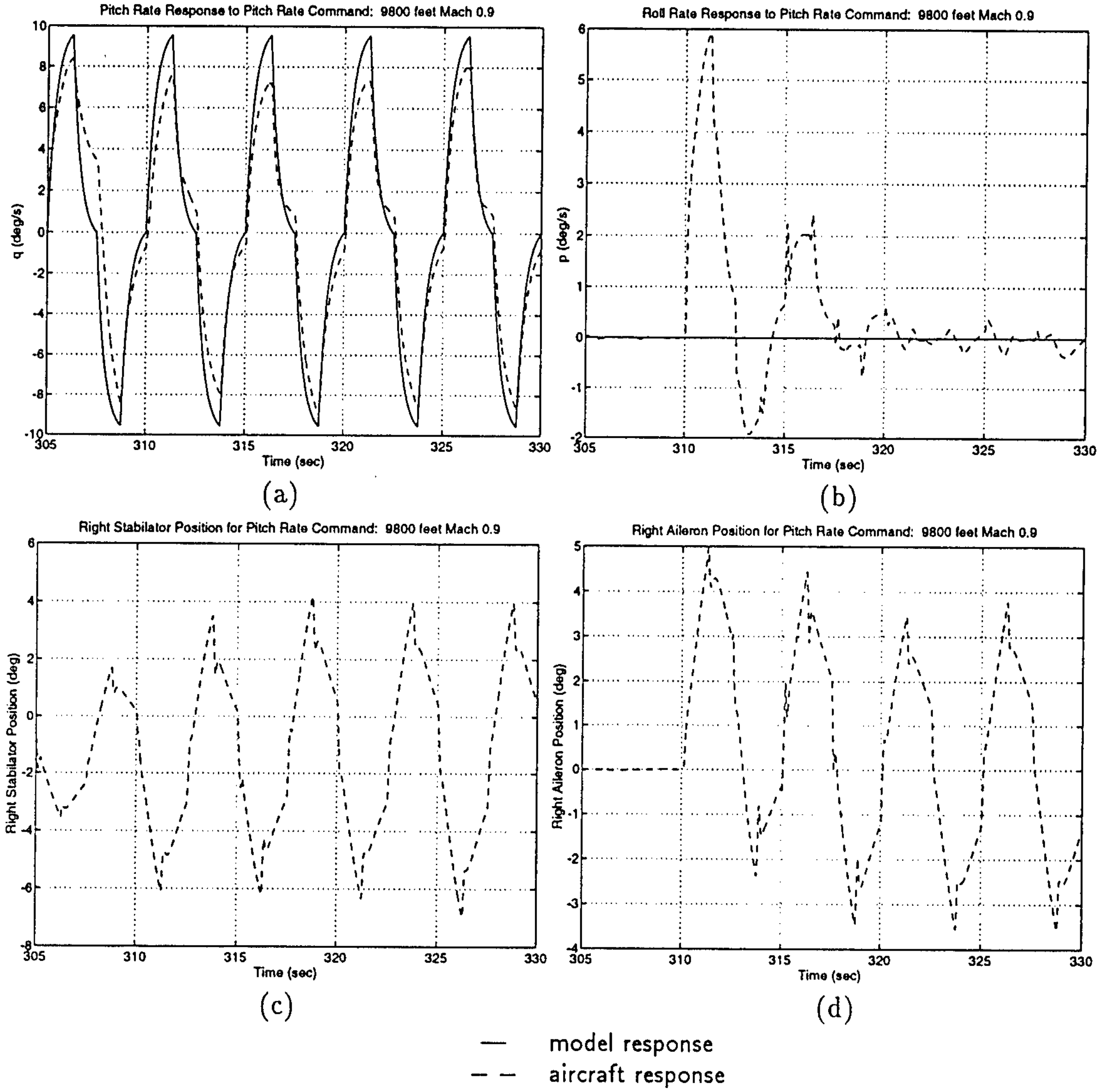
7.1 Flight Condition: 9,800 feet, Mach 0.9

Figure 7.1 illustrates the performance of the aircraft for flight condition 9,800 feet, Mach 0.9, for a locked stabilator. The response of the aircraft is comparable to the previous responses, although the tracking is somewhat degraded. A similar degradation is observed for the 39,800 feet, Mach 1.4 condition, and its discussion is postponed until later.

7.2 Flight Condition: 39,800 feet, Mach 0.6

The 39,800 feet, Mach 0.6 flight condition requires a slight modification to the testing procedure. As mentioned in Section 2.5 and Appendix A, the proportional-integral controller is used to fly the aircraft for the first 305 seconds of flight in order for transients to die out. Certain aircraft parameters, specifically the altitude and airspeed, must be set during this time. For flight conditions well inside the operational envelope, setting the altitude and airspeed to the values specified by the flight condition does not pose a problem using the proportional-integral controller. However, the flight condition of 39,800 feet, Mach 0.6 is close to the boundary of the operational envelope [7]. If the altitude and airspeed are set to the flight condition values in this case, the proportional-integral controller does not maintain flight for 305 seconds. Hence, by 305 seconds the condition of the aircraft is not 39,800 feet, Mach 0.6. The slight modification incorporated in the code was to adjust the set point of the airspeed. The airspeed is set to 678 ft/s (Mach 0.7) for the first 200 seconds of flight and to 631 ft/s (Mach 0.65) afterwards. It may be noted that 631 ft/s does not equal Mach 0.6, but in this case the set point generates an actual aircraft speed of Mach 0.6.

Figure 7.2 shows the response of the aircraft to a locked stabilator failure. The performance of the aircraft is still reasonable but is degraded under this flight condition. Cross-couplings in plot (b) are higher in response to the pitch rate command, and the pitch rate response in plot (a) is delayed. The degradation is related to the extensive rate saturation shown by the linear portions of the right stabilator and aileron positions shown in plots (c) and (d), respectively. At this flight



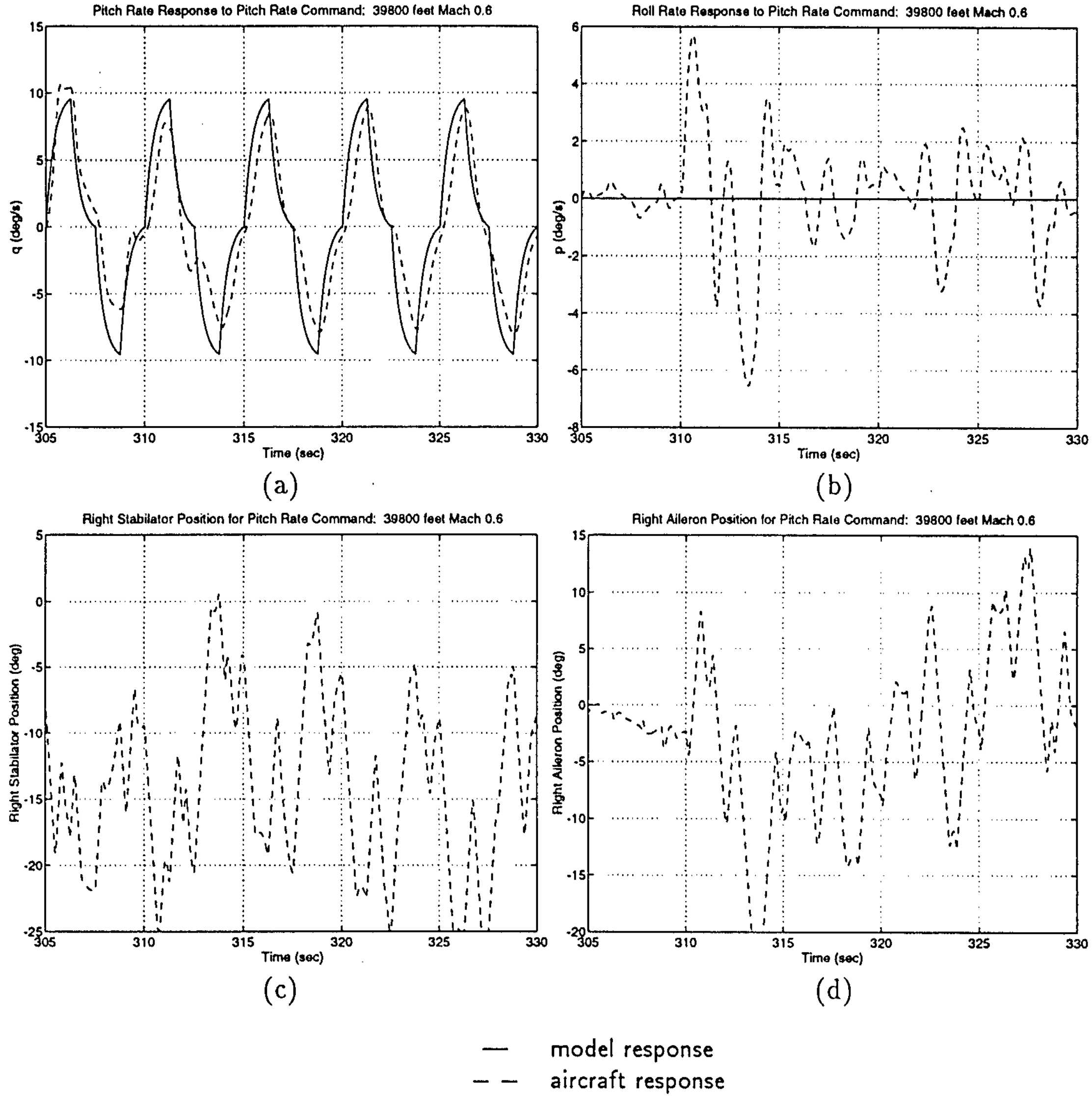
(a)

(b)

(c)

(d)

Figure 7.1: Responses for a Flight Condition of 9,800 feet and Mach 0.9



(a)

(b)

(c)

(d)

Figure 7.2: Responses for a Flight Condition of 39,800 feet and Mach 0.6

condition, the effectiveness of the actuators is much reduced because of the higher altitude and lower air density. This fact can be observed in the state-space matrices found in Appendix A. The $(2,1)$ element of the $\mathbf{B}$ matrix in (2.4) indicates the effectiveness of the stabilator to generate a pitching moment. The greater the magnitude of $\mathbf{B}(2,1)$, the more effective the stabilator is. For the first flight condition, $\mathbf{B}(2,1) = -9.98$. For the flight condition of 39,800 feet, Mach 0.6, $\mathbf{B}(2,1) = -2.88$. The performance is degraded simply because a ±10°/s step is a much more demanding maneuver for this flight condition than the other conditions. Despite this situation, the control system performs well.

7.3 Flight Condition: 39,800 feet, Mach 1.4

Figure 7.3 shows the performance of the aircraft for the supersonic flight condition of 39,800 feet, Mach 1.4 for a step size of ±1°/s with and without integral compensation for a locked stabilator failure. A first observation for this flight condition is that pitch rates are limited, although not because of low pitching authority, but because of low aileron effectiveness. A step size of ±1°/s can maintain low coupling, as shown in plot (b). For step sizes greater than ±1°/s, the ailerons saturate in position. Even though the effectiveness for pitching is much higher at this flight condition than at the nominal condition, the rolling effectiveness is considerably lower, making it difficult to pitch the aircraft without rolling.

Another observation is that tracking of the pitch rate command is not good without integral compensation, although none of the actuators saturate in any way. Simulations of the unfailed aircraft using a BLS identification show a similar behavior with a steady-state value of only ±0.7°/s for a commanded step of ±1°/s. The problem was found to be due to the actuator dynamics, and could be resolved in different ways, one of them being integral compensation. If integral compensation is applied to the unfailed aircraft, the steady-state value of ±1°/s is achieved. Figure 7.3 shows that integral compensation improves the steady-state tracking after the failure.

Unlike the previous flight conditions in which the differential stabilator could be enabled, i.e., by letting $\delta_{Dc} = \frac{5}{3}\delta_{Ac}$, enabling the differential stabilator in the supersonic case yields very poor control of the aircraft. The poor performance is indicated by the state-space matrices. The $\mathbf{B}$ matrix was obtained for the locked stabilator aircraft at the supersonic flight condition with $\delta_{Dc} \neq 0$. $\mathbf{CB}$ is given by

$$\mathbf{CB} = \begin{bmatrix} -6.65 & 5.58 & -0.20 \\ -19.59 & 19.36 & -1.93 \\ -1.47 & 1.22 & -3.45 \end{bmatrix}. \tag{7.1}$$

The first and second columns of $\mathbf{CB}$ are the coefficients of δ_{Hc} and δ_{Ac}, respectively. The approximate equal ratio (≈ 3) between the first two rows in the first and second columns indicates the moments produced for pitch and roll commands are nearly identical in direction. The adaptive algorithm cannot compensate for the failure under this condition. However, with again setting $\delta_{Dc} = 0$, $\mathbf{CB}$ becomes

$$\mathbf{CB} = \begin{bmatrix} -6.90 & 0.03 & -0.10 \\ -20.04 & 2.97 & -1.49 \\ -1.69 & 0.02 & -3.47 \end{bmatrix}. \tag{7.2}$$

Although this matrix is still poorly conditioned, the simulation results were much better with this design.

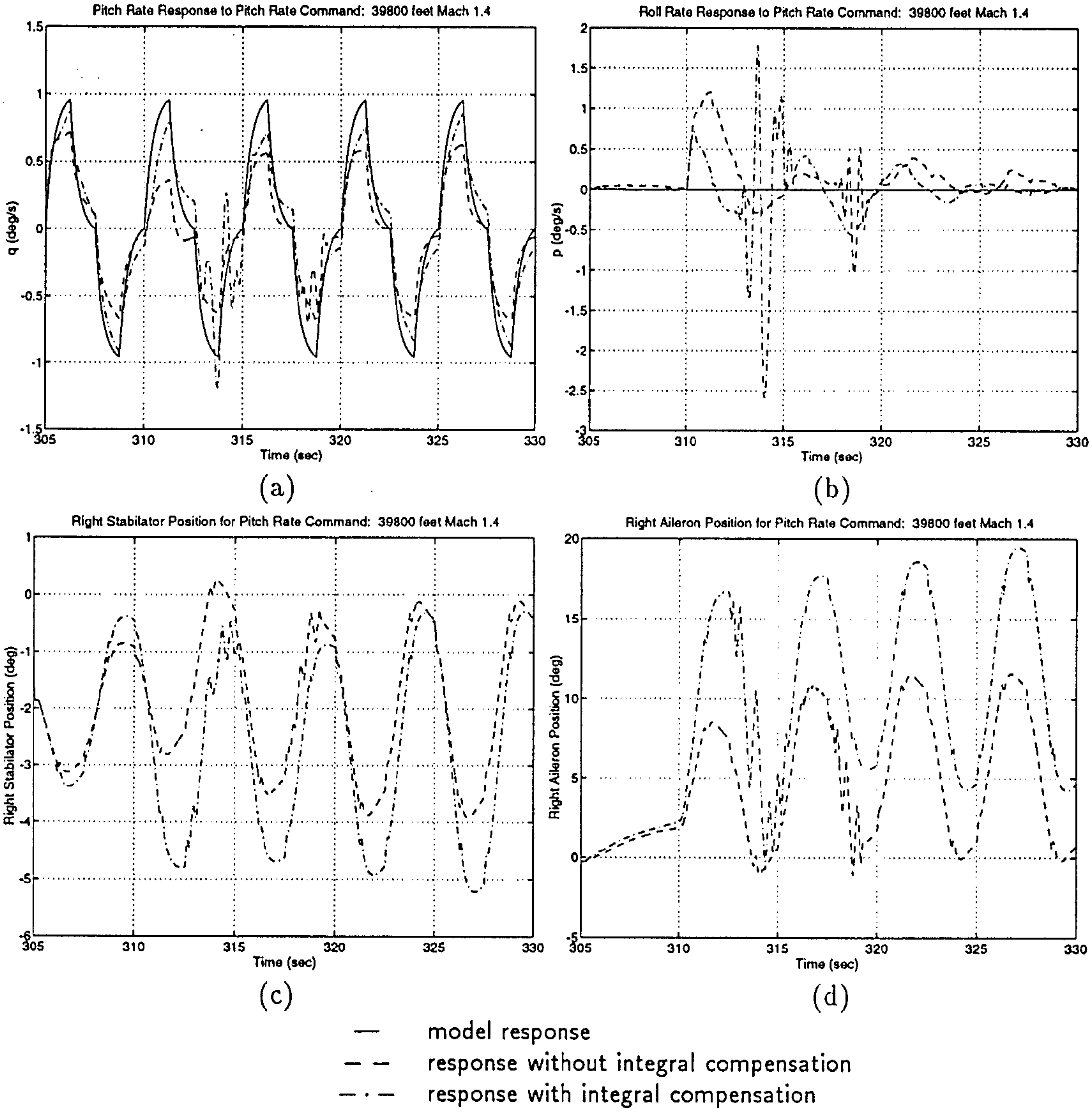


Figure 7.3: Responses for a Flight Condition of 39,800 feet and Mach 1.4

7.4 Locked Aileron Failure

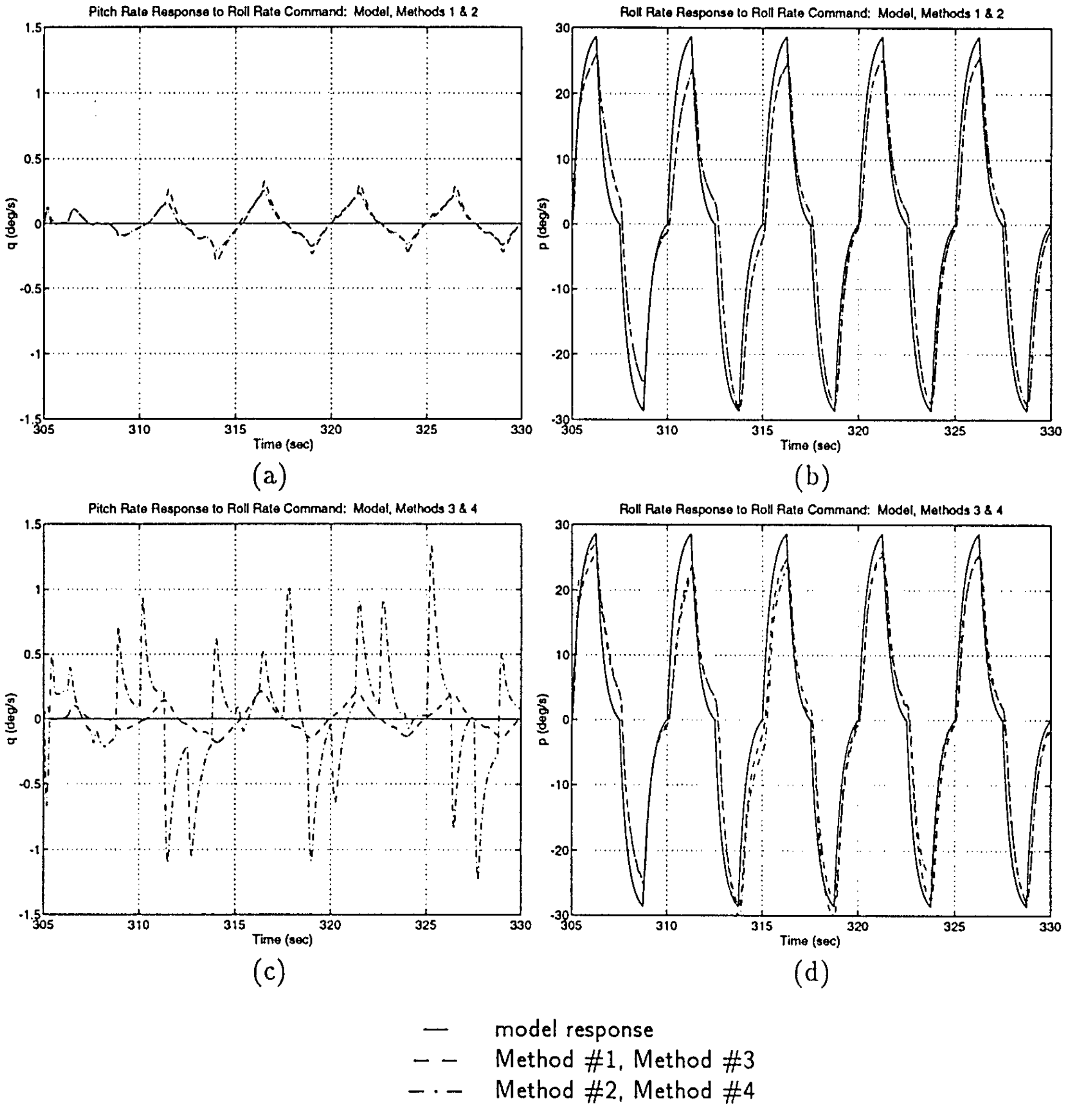
The adaptive algorithm is effective for failures other than the locked stabilator and missing stabilator. A locked aileron failure is considered for illustration. As opposed to the stabilator failures which induce rolling moments, the aileron failure does not induce significant pitching moments, so that this failure is less demanding. Figures 7.4 and 7.5 illustrate the effectiveness of the command limiting techniques to roll rate commands with step sizes of ±30°/s and ±50°/s, respectively. For the responses in Figure 7.4 for ±30°/s, very little difference is found between the techniques. Couplings in plots (a) and (c) remain low, and the response nearly approaches the model in plots (b) and (d). Command limiting was applied about 20% of the time, and the "no ρ" case occurred less than 1% of the time. The responses in Figure 7.5 for ±50°/s do not provide the same conclusions. The command limiting methods perform well, except the least-squares approximation to commanded accelerations, i.e., Method # 4. The least-squares method has an extremely large coupling that develops for a period of time, although the response is recovered as shown in plots (c) and (d). For the step size of ±50°/s, command limiting occurred about 35% of the time, except in the least-squares case in which command limiting occurred about 60% of the time. The significant amount of command limiting can be related to the transient response observed in Figure 7.5. For the relaxation of control requirements and the scaling of reference inputs methods, the "no ρ" case occurred less than 3% of the time.

7.5 Combined Stabilator/Aileron Failure

A combined stabilator and aileron failure was also applied to the aircraft model. In this case, both the left stabilator and left aileron were locked at 310 seconds. Figure 7.6 shows the performance of the aircraft with the combined failure to a pitch rate command of ±10°/s. The control algorithm is able to achieve a satisfactory response. The deviations of both the unfailed stabilator and aileron, shown in plots (c) and (d), respectively, increase significantly due to this failure. The algorithm is also capable of handling a roll rate command of ±30°/s as shown in Figure 7.7.

7.6 Summary

The adaptive algorithm was shown to perform well for other flight conditions and failure types. Degradation in performance at various flight conditions was due to a variety of factors including loss of actuator effectiveness. At one flight condition, actuator dynamics had a significant impact. Further research is needed to address this problem. The adaptive controller has been shown to be capable of controlling the aircraft during a locked aileron and a combined locked stabilator and locked aileron failure. The adaptive algorithm generally performed well for large step sizes in roll rate, with roll rate most significantly affected by an aileron failure. Some interesting behavior that resulted was that the least-squares approximation to commanded accelerations seemed to perform worse for a very large roll rate command with a locked aileron.



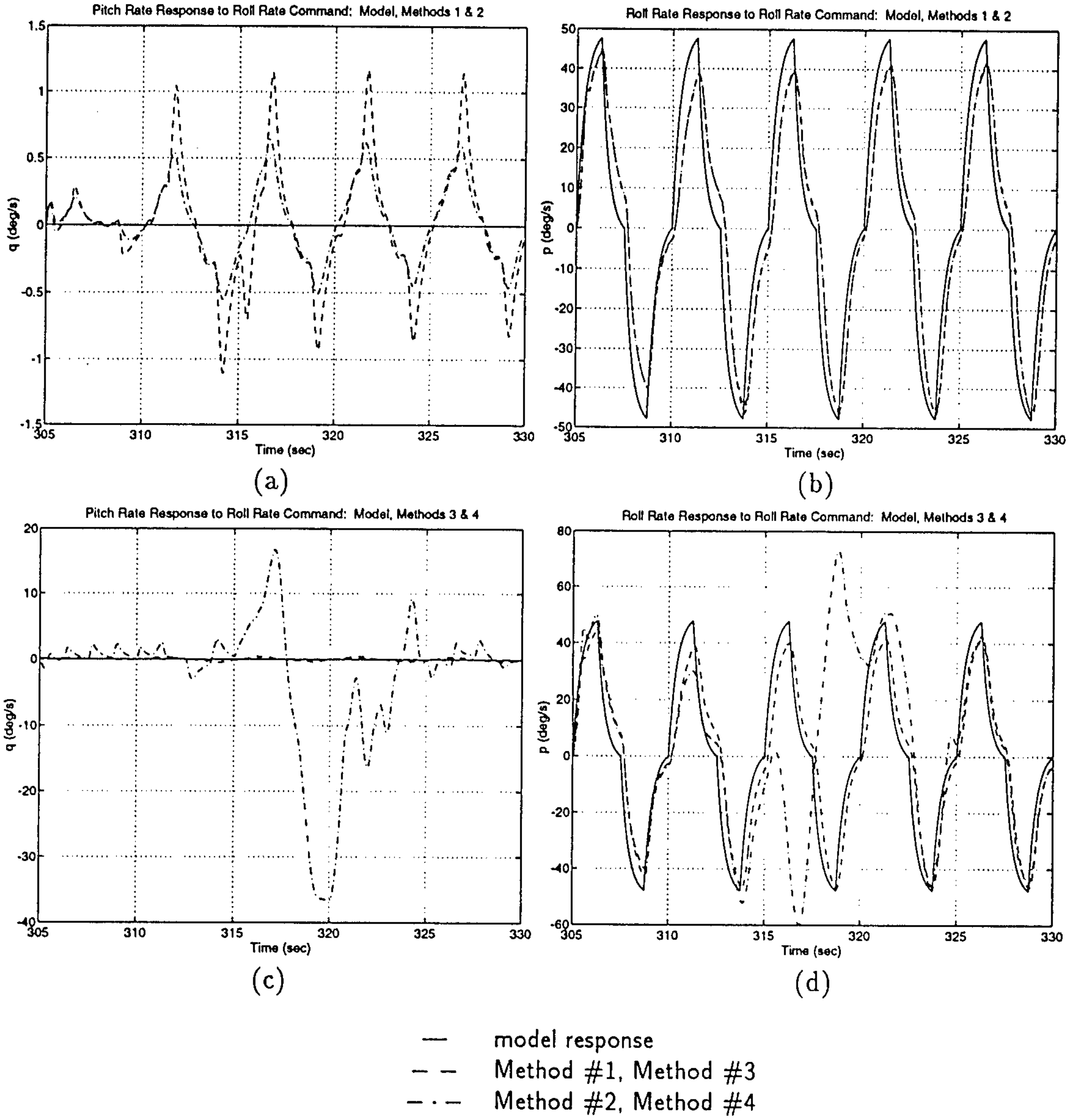
(a)

(b)

(c)

(d)

Figure 7.4: Roll Rate Command Responses for ±30°/s and All Command Limiting Methods with a Locked Aileron Failure



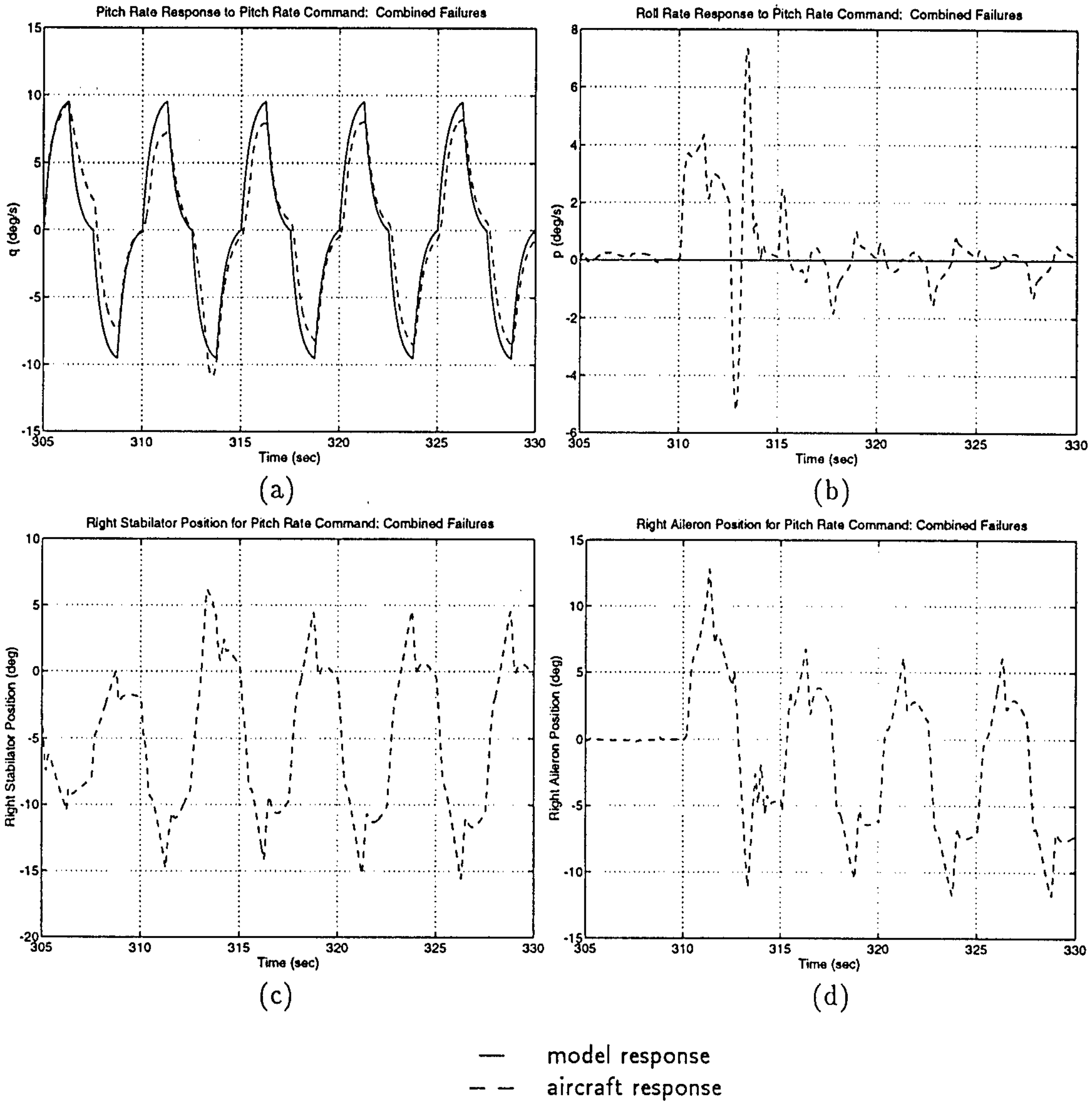
(a)

(b)

(c)

(d)

Figure 7.5: Roll Rate Command Responses for ±50°/s and All Command Limiting Methods with a Locked Aileron Failure



(a)

(b)

(c)

(d)

Figure 7.6: Pitch Rate Command Responses for ±10°/s with a Combined Locked Stabilator/Aileron Failure

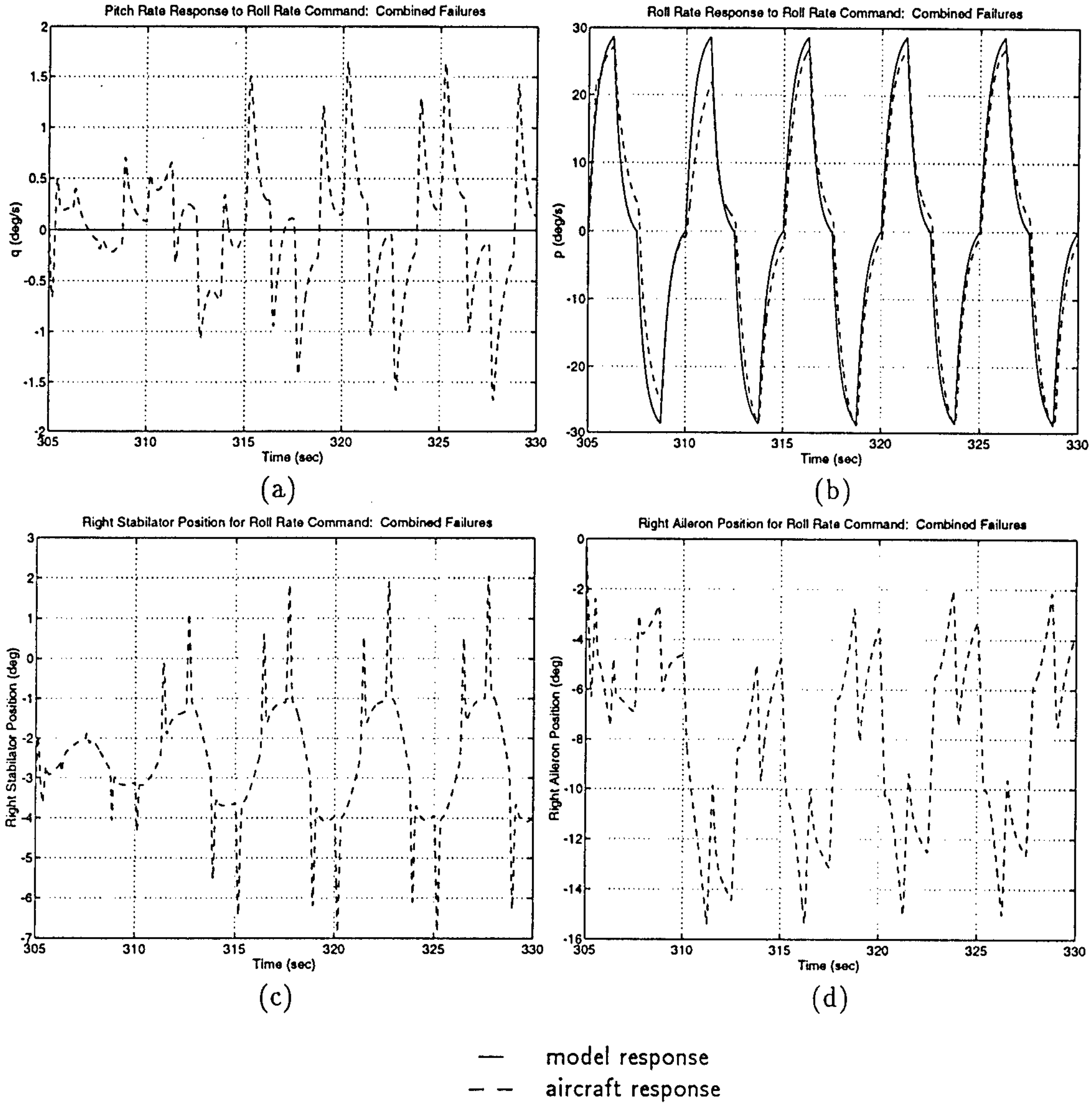


Figure 7.7: Roll Rate Command Responses for ±30°/s with a Combined Locked Stabilator/Aileron Failure

Chapter 8

Miscellaneous Observations

A few observations made in the course of the research were besides the focus of the research. This chapter addresses these issues.

8.1 Slow Divergence

The adaptive controller does not regulate pitch, roll, and yaw angular positions in the control law. Since these values are the integrals of the rates, divergence may easily result. In practice, the pilot would regulate angular position given control of the angular rates. In the simulation, an autopilot may be applied to return the aircraft to nominal values of pitch, roll, and yaw angles and prevent the divergence. The autopilot matches the model (2.8), (2.10), (2.11) and is given by

$$\begin{aligned} \dot{q} &= -2.5q + 2.5q_c, \\ \dot{p} &= -2.5p + 2.5p_c, \\ \dot{r} &= -2.5r + 2.5r_c. \end{aligned} \tag{8.1}$$

Using

$$\begin{aligned} \dot{\theta} &= q, \\ \dot{\phi} &= p, \\ \dot{\psi} &= r, \end{aligned} \tag{8.2}$$

where θ, ϕ, and ψ are the angular positions of pitch, roll, and yaw, respectively, and using the commands

$$\begin{aligned} q_c &= g(\theta_c - \theta), \\ p_c &= g(\phi_c - \phi), \\ r_c &= g(\psi_c - \psi), \end{aligned} \tag{8.3}$$

the autopilot will track the desired angles with the transfer function

$$\frac{2.5g}{s^2 + 2.5s + 2.5g}. \tag{8.4}$$

The constant g is selected to be 1.6 so that the closed-loop poles are located at $-1.25 \pm \jmath 1.56$.

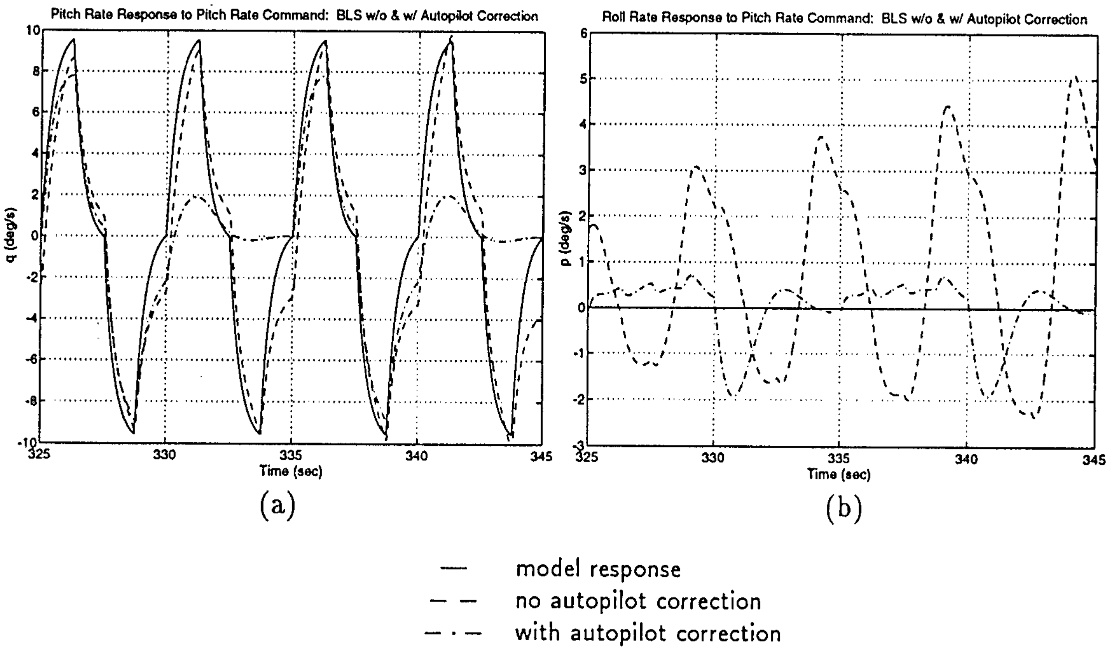


Figure 8.1: Demanding Flight Maneuver Rate Responses with and without Autopilot Correction

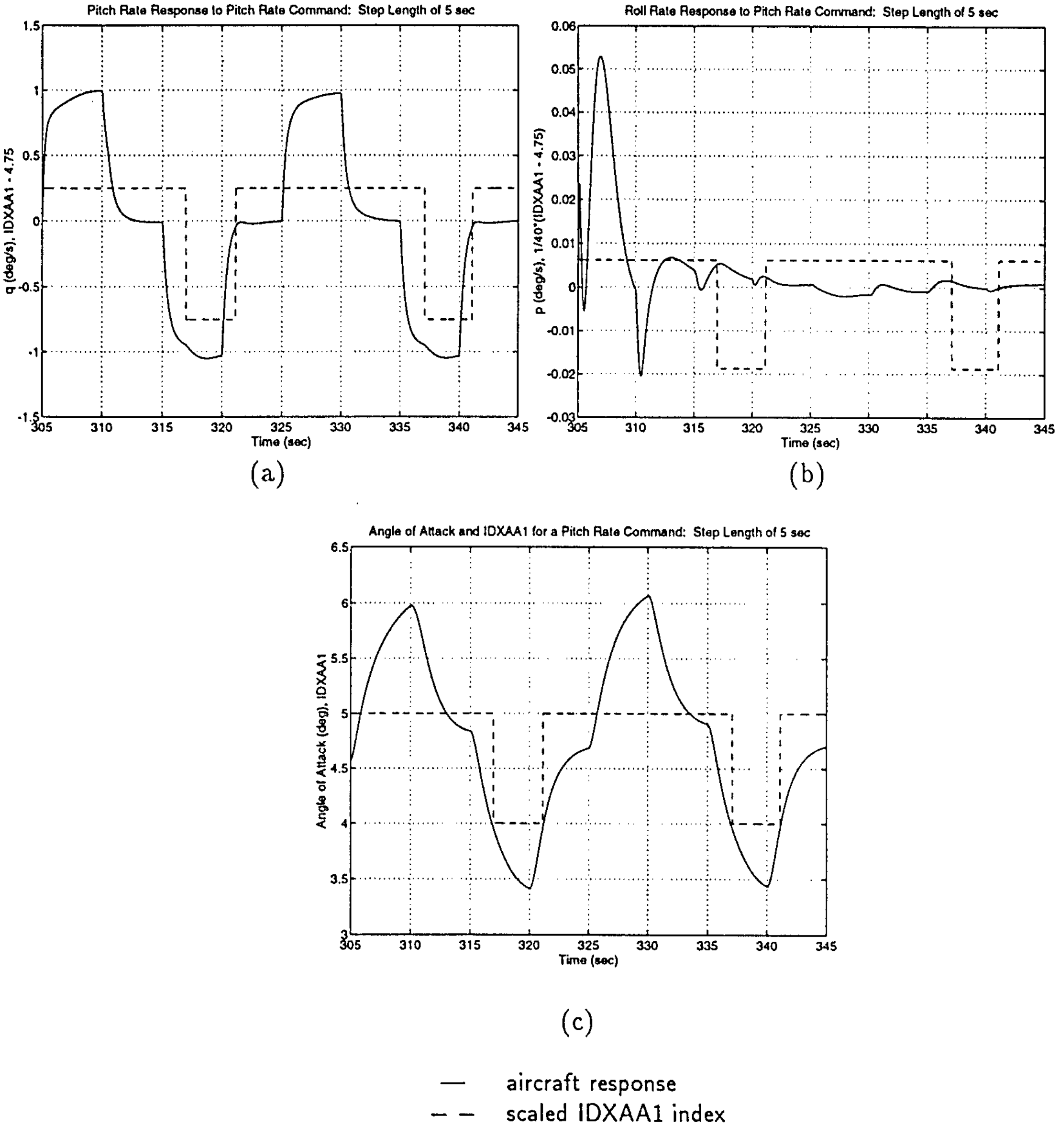
Figure 8.1 compares the unfailed batch least-squares responses with and without applying an autopilot correction in between stepping segments of size ±10°/s without control allocation. Drift in the roll rate response to a pitch rate command shown in plot (b) is observed without the autopilot. Overall, the responses show that slow divergence due to the uncontrolled states can easily be compensated for.

8.2 Glitches in Responses

Some simulations show behavior that is unexpected. Figure 8.2 shows a simulation for the unfailed aircraft where the length of the steps is extended to 5 seconds, as opposed to 1.25 seconds. As the pitch rate response approaches the final value of −1°/s, a kink is observed in the response of plot (a), which makes the response look like a double step response. Such glitches often appear in the simulations and are due to the discrete nature of look-up tables used in the simulations. Specifically, Figure 8.2 shows the index of the look-up tables based on the angle of attack. It is superimposed on the pitch rate, roll rate, and angle of attack plots and is scaled for easy comparison. The index is IDXAA1 and is shown as a dashed line in the figure. IDXAA1 changes for every 4° change in the angle of attack. Viewing plots (a) and (c) of the figure shows that the glitch occurs at a point where the angle of attack index changes. The glitch is not a problem of the controller but an imperfection in the interpolation of the look-up tables.

8.3 Nonlinear Effects of the Angle of Attack

Extending the length of time for the pitch rate command produces a problem for the control law, because of large angles of attack. The standard length of time for a step is 1.25 seconds. Extending the step time length to 2.5 seconds causes a larger excursion in angle of attack. A ±10°/s step in



(a)

(b)

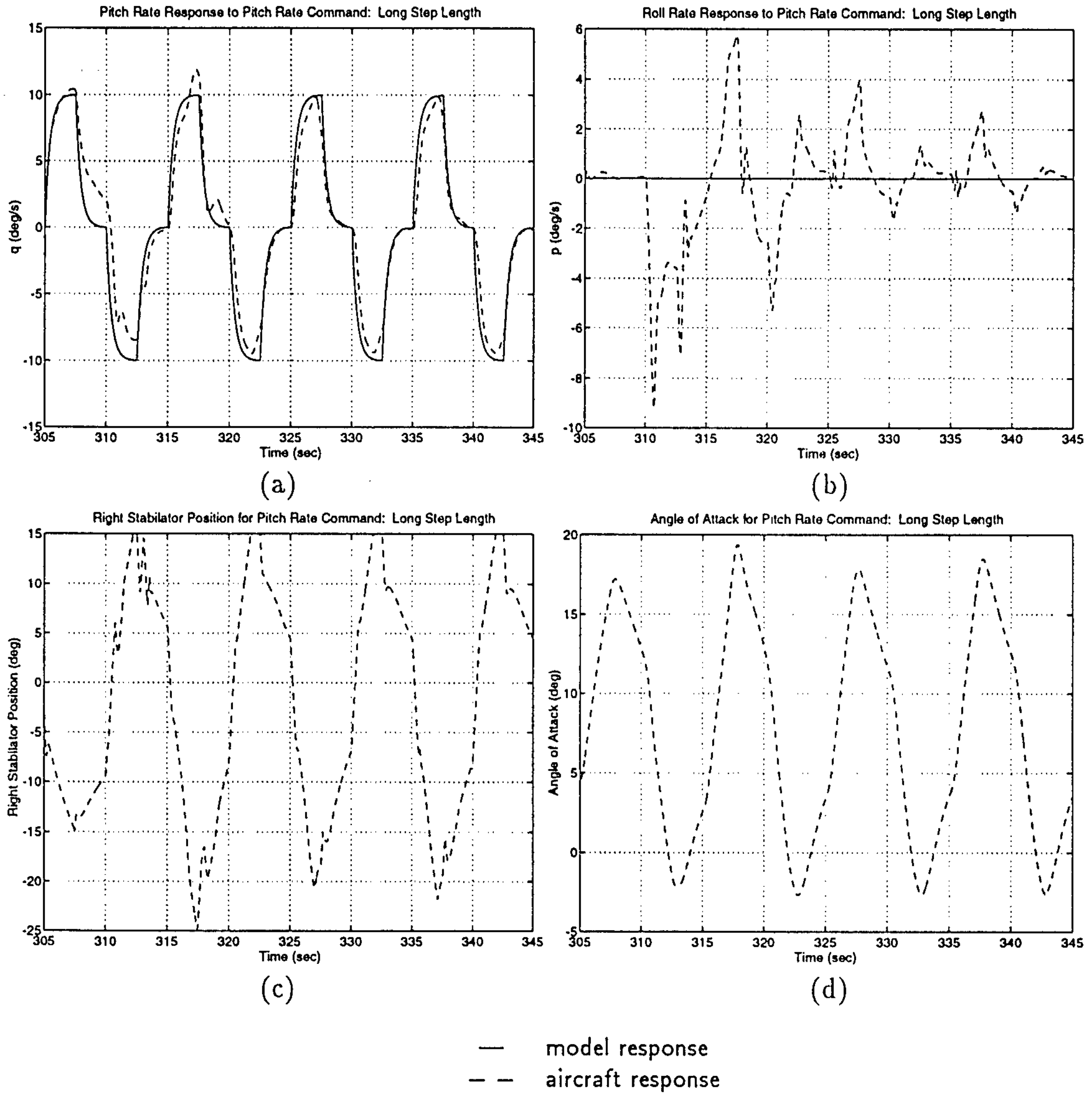
(c)

Figure 8.2: Unfailed Aircraft Responses to a Pitch Rate Command for Step Time Length of 5 sec

pitch causes the angle of attack to change from 4° to about 20° in the 2.5 second interval. Figure 8.3 illustrates that spikes occur in the coupling response in plot (b) when the positive steady-state value in the desired axis is approached. At this time, the angle of attack is approaching 20°. The problem occurring here is that the ailerons begin losing effectiveness rapidly as the angle of attack increases. The solution to this problem is not simple. Modifications to the control law are required and would be the subject of future research in regards to the model reference adaptive controller.

8.4 Summary

Certain interesting behaviors occurred in the simulations that were not directly related to the adaptive controller or were beyond the scope of the research. Responses of the aircraft could become poor due to slow divergence of the angular positions of the aircraft, but this divergence would be corrected by the pilot in practice, or could be eliminated by an autopilot. Another behavior is that of glitches appearing in certain simulations caused by the aircraft simulation software. The most significant problem that may require more research is that of determining corrective actions for the nonlinear effects of the angle of attack.



Pitch Rate Response to Pitch Rate Command: Long Step Length

Roll Rate Response to Pitch Rate Command: Long Step Length

Right Stabilator Position for Pitch Rate Command: Long Step Length

Angle of Attack for Pitch Rate Command: Long Step Length

(a)

(b)

(c)

(d)

Figure 8.3: Responses for Larger Excursions of the Angle of Attack

Chapter 9

Conclusions

The goal of this research was to improve the robustness of an adaptive flight control law. An algorithm was selected that was simple, was capable of handling the cross-couplings of a failed aircraft, and utilized a state-feedback formulation. The basic algorithm combining a model reference control law and a stabilized recursive least-squares identification procedure was shown to be capable of providing control reconfiguration, but showed insufficient robustness properties.

Two alternate identification methods were developed that improved the performance of the control law. The inverse controller formulation was a direct approach that used the same matrices as the control law, but required the calculation of a matrix inverse to determine the parameter matrices. The indirect controller formulation was similar to the inverse controller formulation, with the added benefit of removing the model parameters from the identification. Instability was also eliminated with the filtering of actuator commands and normalization of the algorithm. The use of actuator measurements in the identification slightly improved the results, but this option was not retained because additional fault detection logic would be required.

A significant degradation of performance occurred with larger step size maneuvers, due to saturation in the actuators. To alleviate the problem, four different methods of control allocation were proposed and evaluated. Simulations did not show one particular method to be superior to the others despite the differing levels of complexity of the algorithms. Performance of the adaptive algorithm was significantly improved by all four methods.

Integral compensation was added to the control algorithm to improve tracking of reference inputs. However, integral compensation did not always offer significant improvement to the recursive algorithm because the recursive algorithm was able to fine-tune the controller matrices. In addition, a recursive algorithm combined with integral compensation often caused the parameters to change abruptly, resulting in spiked behavior. The benefits of integral compensation were considered to be limited and possibly not worth the additional complexity.

A brief noise analysis was performed. The results demonstrated that the algorithm was capable of handling Gaussian white noise added to the sensors. Gaussian white noise is not an accurate model of sensor noise, but the results demonstrated that the algorithm was not unduly sensitive to measurement noise.

The adaptive algorithm performed well for other flight conditions and failures. Degradation at some flight conditions was due to reduction of actuator effectiveness. In some other cases, further developments of the algorithm may be needed to improve performance.

Although results of this thesis apply to a specific aircraft model, some of the conclusions should apply to other aircrafts, or even to other multivariable control problems. First, using the modified indirect identification formulation yields a much more stable system than using the direct identification formulation. The indirect identification requires slightly more calculations, i.e., calculating $\mathbf{C}_0$, $\mathbf{G}_0$, and $\mathbf{v}_0$, but the number of parameters that are identified is the same using either the direct or modified indirect formulation. Second, the thesis has shown that rate limiting of the actuators can significantly degrade the performance of the adaptive control law. Ensuring that the control inputs are modified to satisfy the rate constraints brings the performance of the aircraft to within reasonable levels. Third, integral compensation in a control law may be used to minimize steady-state tracking errors, but with less benefits than might be expected since an adaptive algorithm with bias cancellation is able to provide for trim and steady-state tracking. In addition, spiked behavior can occur from the integral compensation.

Appendix A

Aircraft Parameter Matrices

A.1 Off-Line Identification of the Parameter Matrices

The indirect parameter matrices A, B, and d and the direct parameter matrices C_0, G_0, and v_0 are obtained from a run using the proportional-integral controller. The interval for recording data extends for 10 seconds, from 305.0 to 315.0 seconds. Within this time frame, steps are made in h (altitude), ϕ (roll angle), and β (angle of sideslip). The overlapping, but not simultaneous, steps are shown in Table A.1. The lengths and timing of the steps were set arbitrarily with an attempt to provide excitation in all three axes. For identifications involving the base altitude of 39,800 feet, h is set to $39,850.0$ ft, $39,750.0$ ft, and $39,800.0$ ft, respectively in the interval. Data are recorded in the 10 second interval for either the unfailed aircraft or the failed aircraft under any failure type with the failure occurring at 305.0 seconds.

The identification techniques are the same using the inverse identification formulation and the indirect identification formulation described in Section 3.1. Differences appear in the definitions of $\mathrm{w}[n]$ and $\mathrm{z}[n]$.

A.2 State-Space Matrices

For the indirect identification method, the variables which comprise the $\mathrm{w}[n]$ vector, given by

$$\mathrm{w}[n] = \begin{bmatrix} \alpha & q & \beta & p & r & \delta_{Hc} & \delta_{Ac} & \delta_{Rc} & 1 \end{bmatrix}^T, \tag{A.1}$$

and $\mathrm{z}[n]$, given by

$$\mathrm{z}[n] = \begin{bmatrix} \dot{\alpha} & \dot{q} & \dot{\beta} & \dot{p} & \dot{r} \end{bmatrix}^T, \tag{A.2}$$

are recorded during the 10 second interval. A least-squares identification is performed where a matrix θ is produced and is given by

$$\theta = (\mathrm{z}\mathrm{w}^T)(\mathrm{w}\mathrm{w}^T)^{-1} \tag{A.3}$$

where δ_{Hc}, δ_{Ac}, and δ_{Rc} in each $\mathrm{w}[n]$ have been filtered as described in Section 3.2. The resulting $n \times (n+m+1)$ θ matrix contains the sub-matrices A, B, and d.

Tables A.2 and A.3 give the state-space matrices A, B, and d for the unfailed and locked stabilator aircraft with and without differential stabilator movement, respectively, for multiple flight conditions. Flight characteristics for the unfailed aircraft are given in Table A.4.

Table A.1: Identification Stepping Sequence

Variable	Initial Time	Final Time	Set Value
h	305.0	307.5	9850.0 ft
h	307.5	309.5	9750.0 ft
h	309.5	315.0	9800.0 ft
ϕ	305.0	308.0	0.0°
ϕ	308.0	310.0	10.0°
ϕ	310.0	312.0	−10.0°
ϕ	312.0	315.0	0.0°
β	305.0	310.5	0.0°
β	310.5	312.5	15.0°
β	312.5	314.5	−15.0°
β	314.5	315.0	0.0°

A.3 Controller Matrices

The controller matrices $\mathbf{C}_0$, $\mathbf{G}_0$ and $\mathbf{v}_0$ are obtained using (2.15) and the $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{d}$ matrices calculated as in Appendix A.2. $\mathbf{A}_M$ and $\mathbf{B}_M$ are defined by (2.10) and (2.11), respectively. $\mathbf{C}$ is defined as

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.4}$$

in order to define $\mathbf{y}[n] = \begin{bmatrix} q & p & r \end{bmatrix}^T$.

Tables A.5 and A.6 give the controller matrices $\mathbf{C}_0$, $\mathbf{G}_0$, and $\mathbf{v}_0$ for the unfailed and locked stabilator aircraft with and without differential stabilator movement, respectively, for multiple flight conditions. The matrices are obtained from the state-space matrices given in Tables A.2 and A.3.

Table A.2: State-Space Matrices with $\delta_{Dc} = \frac{5}{3}\delta_{Ac}$

	A					*B*			*d*
	-0.7508	1.0075	0.0336	0.0021	-0.0016	-0.0547	-0.0033	-0.0092	3.2762
	-6.2863	-2.9689	0.0628	-0.0296	-0.0377	-10.0003	-0.0770	-0.0007	0.0028
a	-0.0879	0.1550	0.0550	0.0873	-0.9556	0.2729	-0.1610	-0.0879	1.1890
	-0.1990	0.7372	-29.8146	-2.4781	0.7497	1.3651	28.1905	-1.1677	4.8621
	-0.0469	0.0214	4.8857	-0.0913	-0.6239	0.0638	2.2323	-2.9356	0.2620
	-1.5678	1.0160	0.0037	0.0001	-0.0020	-0.0699	0.0049	0.0015	1.9131
	-14.7019	-3.7166	0.8112	0.0486	-0.0467	-32.8240	-0.2883	-0.1741	-0.2540
b	0.0743	-0.0058	0.0358	0.0407	-0.9770	-0.0412	-0.1824	-0.1568	-0.1176
	0.7954	0.2274	-80.4293	-3.7356	-0.0576	0.9372	77.1684	-3.2085	-0.4437
	-0.0977	-0.0242	21.0793	-0.0376	-1.0198	-0.1469	6.3591	-8.8271	-0.2963
	-0.2452	0.9774	0.0180	-0.0014	0.0333	-0.0560	-0.0141	0.0059	2.3869
	-2.0050	-0.9700	-0.0587	-0.0158	-0.0733	-2.9526	0.0320	-0.0063	2.1026
c	-0.0559	-0.0313	-0.0658	0.1876	-0.8681	0.0043	-0.1184	0.0311	0.6489
	-0.2222	0.0317	-12.4265	-0.8319	0.8948	-0.0977	8.9901	0.1320	1.8059
	-0.0238	-0.0370	1.2851	-0.0023	-0.2283	-0.0197	0.8624	-1.1087	-0.1357
	-0.5001	1.0074	0.0312	-0.0001	-0.0030	0.0009	0.0026	-0.0065	1.2175
	-33.3036	-1.3974	0.1806	0.0123	0.0054	-13.4399	-0.2087	0.0073	60.5455
d	-0.0619	0.0232	0.0818	0.0489	-0.9739	0.0481	-0.1375	-0.0625	0.2132
	0.4093	-0.1004	-33.6216	-1.9518	-0.4461	0.5986	34.3778	-1.1398	-0.0255
	0.0636	-0.0230	7.6760	-0.0075	-0.3380	0.0301	2.2882	-3.3631	-0.6400
	-0.7814	1.0042	0.0291	0.0016	-0.0017	-0.0409	0.0254	-0.0021	3.4573
	-6.2451	-2.8557	0.1348	0.0151	-0.0239	-4.7125	3.8476	0.0849	14.9862
e	-0.7220	-0.4524	0.7348	0.1720	-0.8034	-0.1842	-0.1253	0.0265	2.7873
	-1.2155	-2.0952	-27.8080	-2.2300	0.9030	-11.6425	19.5653	-1.1683	-27.6917
	0.1684	-0.1269	4.8959	-0.0806	-0.6605	-1.2966	1.2226	-2.9892	-4.6260
	-1.6000	1.0068	0.0511	0.0041	0.0118	-0.0487	0.0224	0.0067	1.9663
	-14.9011	-3.9323	1.6175	0.0969	0.1638	-15.8540	12.7086	0.0519	9.4532
f	-0.1703	-0.0607	0.1085	0.0433	-0.9460	0.0439	-0.1208	-0.0049	0.2546
	-3.4287	-2.6351	-75.5575	-3.2466	1.4428	-36.1258	49.8657	-2.5302	-15.9522
	0.7466	-0.1439	21.1097	-0.0056	-1.0685	-3.6378	3.5087	-9.5106	-3.3544
	-0.2624	0.9720	0.0305	0.0037	0.0067	-0.0305	0.0117	-0.0022	2.7555
	-1.9904	-0.8958	0.0442	-0.0116	-0.0377	-1.4455	1.2153	0.0104	12.2972
g	-0.2042	-0.2459	0.3506	0.2311	-0.8355	-0.0160	-0.0424	0.0032	2.1446
	-0.1625	-0.5501	-11.4344	-0.7472	0.7148	-3.8610	5.5991	-0.1008	-24.7621
	0.0447	-0.1077	1.3951	0.0219	-0.3389	-0.5619	0.4374	-1.1484	-4.6717
	-0.5013	1.0122	0.0299	-0.0003	-0.0020	-0.0001	-0.0021	-0.0028	1.2202
	-32.0380	-1.6217	0.1277	-0.0211	-0.2009	-6.6475	5.5810	-0.2001	67.6165
h	-0.3457	0.0065	0.2079	0.0666	-0.9229	0.1191	-0.0850	-0.0714	1.0227
	-2.4882	-0.5936	-32.4632	-1.8853	-0.3089	-19.5900	19.3566	-1.9338	-23.2986
	-0.5148	-0.0492	7.8886	0.0027	-0.2886	-1.4782	1.2176	-3.4458	-1.4999

a	-	9800 ft	Mach 0.5	unfailed	e	-	9800 ft	Mach 0.5	locked
b	-	9800 ft	Mach 0.9	unfailed	f	-	9800 ft	Mach 0.9	locked
c	-	39800 ft	Mach 0.6	unfailed	g	-	39800 ft	Mach 0.6	locked
d	-	39800 ft	Mach 1.4	unfailed	h	-	39800 ft	Mach 1.4	locked

Table A.3: State-Space Matrices with $\delta_{D_c} = 0$

	A					*B*			*d*
a	-0.7599	1.0026	0.0244	0.0002	-0.0021	-0.0700	0.0002	-0.0047	3.2790
	-6.2749	-2.9616	0.1042	-0.0263	0.0180	-9.9846	-0.1242	0.0179	-0.0066
	-0.0865	0.1007	-0.0902	0.0877	-1.1007	0.1608	-0.0913	-0.0611	0.8685
	-0.1270	0.4762	-30.2641	-2.4635	0.5040	0.8894	10.7071	-1.6015	3.1396
	-0.0331	0.0230	4.8450	-0.0907	-0.6357	0.0473	0.1660	-3.0015	0.1418
b	-1.5731	1.0186	-0.0104	-0.0030	-0.0048	-0.0688	0.0059	0.0014	1.9234
	-14.7169	-3.7149	0.7575	0.0538	-0.0114	-32.8112	-0.0703	-0.0710	-0.2282
	0.0387	-0.0120	0.0630	0.0440	-1.0077	-0.0631	-0.0922	-0.1152	-0.0735
	0.3621	0.0854	-84.4544	-3.8654	-0.2023	0.4927	22.5359	-3.6291	-0.2469
	-0.0457	-0.0217	20.3607	-0.0609	-1.0410	-0.0938	0.4239	-8.9232	-0.3524
c	-0.2514	0.9975	0.0328	-0.0001	0.0544	-0.0399	0.0089	0.0111	2.5802
	-1.9943	-0.9338	-0.2792	-0.0302	-0.1843	-2.8756	-0.0149	0.0210	2.5012
	-0.0985	0.1124	-0.1928	0.2147	-1.3929	0.1640	-0.0245	-0.0073	1.9538
	-0.4102	0.3859	-12.4031	-0.7302	0.0991	0.3119	2.7321	-0.1330	6.5433
	-0.0503	0.0371	1.2192	0.0150	-0.3652	0.0196	0.0140	-1.1276	0.3918
d	-0.4962	1.0167	0.0134	-0.0045	0.0044	0.0190	-0.0103	0.0070	1.2472
	-33.0930	-1.3887	0.3751	0.0251	0.1680	-13.3610	-0.0379	0.0732	60.1684
	-0.0008	0.0347	0.0596	0.0501	-1.0181	0.0619	-0.0561	0.0215	0.0269
	-0.5169	-0.0246	-33.0118	-1.8412	0.0058	-0.1840	2.8938	-1.3813	0.9318
	0.0544	0.0021	7.1899	-0.0474	-0.5679	0.0461	0.0343	-3.5316	-0.6290
e	-0.7566	1.0146	0.0059	-0.0015	-0.0098	-0.0294	0.0020	-0.0038	3.3819
	-6.3823	-2.8453	0.0109	-0.0112	-0.0141	-4.9151	-0.0154	-0.0038	15.0893
	-0.0449	-0.1370	0.3714	0.1396	-0.9621	0.1197	-0.1602	-0.0781	0.5566
	0.1352	-0.1132	-29.2927	-2.4059	0.7737	-10.5237	10.4894	-1.5874	-30.8504
	-0.0502	-0.0388	4.9104	-0.0886	-0.6048	-1.3384	0.1439	-2.9833	-3.7653
f	-1.5705	1.0129	0.0051	-0.0020	0.0005	-0.0486	0.0010	0.0029	1.9337
	-14.8281	-3.7516	0.4363	0.0324	-0.0101	-16.4431	-0.0161	-0.0501	9.1773
	0.0086	-0.0545	-0.1006	0.0327	-1.0424	-0.0042	-0.0638	-0.1013	0.0034
	-0.1417	-0.0875	-85.8416	-3.9473	-0.2669	-35.1505	22.4301	-3.7710	-19.5522
	0.0244	0.2116	19.9857	-0.0808	-1.0728	-3.8319	0.4734	-8.9697	-2.5368
g	-0.2671	0.9858	0.0387	0.0010	0.0366	-0.0229	0.0035	0.0075	2.8774
	-1.9202	-0.8826	-0.1273	-0.0187	-0.1196	-1.4418	0.0101	0.0032	11.7995
	-0.0350	-0.0600	0.2628	0.2281	-0.8504	0.1000	-0.0425	-0.0127	0.8804
	-0.1382	0.0741	-11.7403	-0.7709	0.8784	-3.5560	2.5357	-0.0909	-23.7294
	-0.0097	-0.0539	1.2805	0.0087	-0.2809	-0.5613	-0.0157	-1.1260	-4.1603
h	-0.4989	1.0168	0.0552	-0.0009	0.0037	0.0183	-0.0066	0.0023	1.2489
	-32.5991	-1.5689	-0.2683	-0.0286	-0.1756	-6.8991	0.0337	-0.0960	69.0377
	-0.6259	0.0887	0.5004	0.1013	-0.8338	0.3101	-0.0138	0.0287	1.9309
	-5.4085	-0.4432	-33.0980	-1.8378	-0.1025	-20.0382	2.9741	-1.4917	-18.2700
	-0.1756	-0.1540	7.4708	-0.0218	-0.4275	-1.6852	0.0170	-3.4701	-2.7667

a	-	9800 ft	Mach 0.5	unfailed	e	-	9800 ft	Mach 0.5	locked
b	-	9800 ft	Mach 0.9	unfailed	f	-	9800 ft	Mach 0.9	locked
c	-	39800 ft	Mach 0.6	unfailed	g	-	39800 ft	Mach 0.6	locked
d	-	39800 ft	Mach 1.4	unfailed	h	-	39800 ft	Mach 1.4	locked

Table A.4: Flight Condition Characteristics

Altitude h (ft)	Speed M (Mach)	Airspeed v (ft/s)	Air Density ρ (10^{-3} slug/ft)	Trim α α_{trim} (°)	Trim δ_H $\delta_{H\text{trim}}$ (°)
9,800	0.5	539	1.8	4.7	−2.9
9,800	0.9	970	1.8	1.3	−0.6
39,800	0.6	581	0.59	11.4	−7.0
39,800	1.4	1355	0.59	2.5	−1.6

Table A.5: Controller Matrices with $\delta_{Dc} = \frac{5}{3}\delta_{Ac}$

	C_0			G_0					v_0
a	-0.2501	-0.0007	0.0003	-0.6289	-0.0467	-0.0029	-0.0029	-0.0038	0.0016
	0.0123	0.0916	-0.0364	0.0375	-0.0244	1.1633	-0.0020	0.0001	-0.1743
	0.0039	0.0696	-0.8793	-0.0012	-0.0123	2.5488	-0.0327	0.6391	-0.0433
b	-0.0762	-0.0004	0.0017	-0.4478	-0.0370	-0.0028	0.0013	-0.0024	-0.0076
	0.0010	0.0334	-0.0122	-0.0052	-0.0027	1.1768	0.0163	0.0080	0.0046
	0.0020	0.0241	-0.2920	-0.0073	-0.0040	3.2359	0.0075	0.1735	-0.0301
c	-0.8468	0.0025	0.0051	-0.6789	0.5183	-0.0100	-0.0070	-0.0304	0.7107
	-0.0093	0.2750	0.0328	0.0173	0.0027	1.3497	-0.1835	-0.1285	-0.1890
	0.0078	0.2138	-2.2294	0.0041	-0.0405	2.2091	-0.1447	1.9496	-0.2820
d	-0.1861	-0.0011	-0.0000	-2.4785	0.0820	-0.0017	0.0012	0.0002	4.5061
	0.0033	0.0744	-0.0252	0.0319	0.0013	1.0780	-0.0164	0.0351	-0.0846
	0.0006	0.0506	-0.7605	0.0184	-0.0052	3.0158	-0.0134	0.6667	-0.2076
e	-1.0063	0.2047	-0.1086	-2.4069	0.0229	2.5436	-0.0195	-0.0037	8.0984
	-0.5871	0.2504	-0.1145	-1.3371	0.1205	3.0411	-0.0272	-0.0118	6.0807
	0.1964	0.0136	-0.8361	0.5536	-0.0031	1.7783	-0.0296	0.6121	-2.5735
f	-0.3715	0.0966	-0.0277	-2.0737	-0.1126	3.3953	0.0432	-0.0155	1.9843
	-0.2669	0.1205	-0.0335	-1.4158	-0.0278	4.0984	0.0463	-0.0329	1.7335
	0.0436	0.0075	-0.2646	0.3493	0.0177	2.4329	-0.0000	0.1443	-0.4722
g	-4.0665	0.8917	-0.1153	-3.1775	2.8006	4.2146	-0.6431	-0.2166	28.6192
	-2.7874	1.0608	-0.1185	-2.1481	2.0169	4.9672	-0.7557	-0.2429	23.9967
	0.9281	-0.0323	-2.1657	0.7755	-0.6959	1.0446	0.0459	1.8954	-8.9315
h	-2.4374	0.7192	-0.2621	-30.5737	1.0219	10.2910	-0.1971	0.1248	72.4688
	-2.4488	0.8565	-0.3384	-30.5988	1.0570	12.3147	-0.2309	0.2084	74.0099
	0.1803	-0.0059	-0.7327	2.1535	-0.0792	2.2263	0.0038	0.6619	-5.3703

a	-	9800 ft	Mach 0.5	unfailed	e	-	9800 ft	Mach 0.5	locked
b	-	9800 ft	Mach 0.9	unfailed	f	-	9800 ft	Mach 0.9	locked
c	-	39800 ft	Mach 0.6	unfailed	g	-	39800 ft	Mach 0.6	locked
d	-	39800 ft	Mach 1.4	unfailed	h	-	39800 ft	Mach 1.4	locked

Table A.6: Controller Matrices with $\delta_{D_c} = 0$

	C_0			G_0					v_0
a	-0.2506	-0.0029	0.0001	-0.6293	-0.0457	-0.0249	-0.0026	0.0024	0.0030
	0.0204	0.2357	-0.1256	0.0615	-0.0400	3.0956	-0.0078	0.0460	-0.2888
	-0.0028	0.0130	-0.8399	-0.0175	0.0047	1.7850	-0.0307	0.6237	0.0313
b	-0.0762	-0.0003	0.0007	-0.4485	-0.0370	0.0088	0.0015	-0.0008	-0.0069
	0.0018	0.1118	-0.0455	-0.0064	-0.0033	4.1465	0.0599	0.0356	0.0048
	0.0009	0.0053	-0.2823	-0.0007	-0.0022	2.4786	-0.0040	0.1652	-0.0392
c	-0.8700	-0.0047	-0.0156	-0.6951	0.5460	-0.1127	-0.0071	-0.0506	0.8851
	0.0986	0.9161	-0.1062	0.2269	-0.2016	4.6079	-0.6467	0.0617	-2.4798
	-0.0139	0.0113	-2.2186	-0.0539	0.0399	1.1366	0.0051	1.8930	0.3319
d	-0.1871	-0.0024	-0.0029	-2.4770	0.0831	0.0046	0.0025	0.0148	4.5029
	-0.0131	0.8678	-0.3397	0.0131	0.0146	12.4377	-0.2350	0.2614	-0.0931
	-0.0026	0.0084	-0.7112	-0.0168	0.0018	2.1569	-0.0157	0.5498	-0.1202
e	-0.5073	-0.0008	0.0011	-1.2950	-0.0701	-0.0088	-0.0022	-0.0034	3.0542
	-0.4780	0.2394	-0.1268	-1.2358	-0.0571	3.0558	-0.0156	0.0193	5.6481
	0.2045	0.0119	-0.8446	0.5046	0.0157	1.7972	-0.0295	0.6377	-2.3599
f	-0.1520	-0.0001	0.0009	-0.9014	-0.0762	0.0149	0.0019	-0.0011	0.5578
	-0.2293	0.1123	-0.0459	-1.3531	-0.1070	4.2618	0.0665	0.0373	1.6731
	0.0528	0.0060	-0.2815	0.3164	0.0505	2.4467	-0.0063	0.1616	-0.4328
g	-1.7488	0.0069	-0.0056	-1.3428	1.1311	-0.0537	-0.0178	-0.0812	8.3106
	-2.4201	0.9950	-0.0872	-1.8041	1.5343	4.5941	-0.7060	-0.3880	20.7214
	0.9055	-0.0173	-2.2163	0.6859	-0.6331	1.0998	0.0265	2.0167	-8.1265
h	-0.3767	0.0042	0.0086	-4.9026	0.1416	-0.0103	-0.0053	-0.0334	10.4436
	-2.4525	0.8701	-0.3062	-30.1187	1.0488	12.1718	-0.2612	0.1172	73.7459
	0.1709	0.0022	-0.7261	2.1824	-0.1080	2.2176	-0.0050	0.6140	-5.5069

a	-	9800 ft	Mach 0.5	unfailed	e	-	9800 ft	Mach 0.5	locked
b	-	9800 ft	Mach 0.9	unfailed	f	-	9800 ft	Mach 0.9	locked
c	-	39800 ft	Mach 0.6	unfailed	g	-	39800 ft	Mach 0.6	locked
d	-	39800 ft	Mach 1.4	unfailed	h	-	39800 ft	Mach 1.4	locked

Appendix B

Calculation of the Local Angle of Attack for Missing Stabilator

The local angle of attack for a missing stabilator is calculated using the following procedure. First, the *rate of change of downwash angle with angle of attack* is calculated by the following equation (see [23]):

$$\frac{d\epsilon}{d\alpha} = \frac{2}{\pi AR} C_{L\alpha}. \tag{B.1}$$

Using the parameters from Table 2.1 of $S = 608$ ft^2 and $b = 42.8$ ft, the aspect ratio is calculated as

$$AR = \frac{b^2}{s} = \frac{(42.8 \text{ ft})^2}{608 \text{ ft}^2} = 3.01. \tag{B.2}$$

Table B.1 lists the coefficients of lift for various values of angle of attack (α) and Mach number (M) obtained from the tables of the simulation code. The table values are interpolated for the condition of Mach 0.5. For an 8° change in α,

$$\Delta C_L = 0.50 - (-0.03) = 0.53 \text{ for Mach } 0.4 \tag{B.3}$$

$$\Delta C_L = 0.52 - (-0.05) = 0.57 \text{ for Mach } 0.6 \tag{B.4}$$

Since the flight condition of Mach 0.5 is desired, average the values of ΔC_L from (B.3) and (B.4) to obtain $\Delta C_L = 0.55$. Calculate

$$\frac{\Delta C_L}{\Delta\alpha} = \frac{0.55}{8}\frac{180}{\pi} = 3.94 = C_{L\alpha}. \tag{B.5}$$

Table B.1: Coefficients of Lift for Mach 0.4 and 0.6

α (°)	Mach 0.4	Mach 0.6
0	−0.03	−0.05
4	0.24	0.23
8	0.50	0.52

Substituting the values for AR and $C_{L\alpha}$ in (B.1) yields

$$\frac{d\epsilon}{d\alpha} = \frac{2}{\pi 3.01} 3.94 = 0.83. \tag{B.6}$$

The *downwash angle* ϵ can be calculated as

$$\epsilon = \epsilon_0 + \frac{d\epsilon}{d\alpha}\alpha. \tag{B.7}$$

The local angle of attack at the tail α_L is then

$$\begin{aligned} \alpha_L &= \alpha - \epsilon \\ &= \alpha - \epsilon_0 - 0.83\alpha \\ &= 0.17\alpha - \epsilon_0 \end{aligned} \tag{B.8}$$

Unfortunately, the constant ϵ_0 cannot be determined easily, even using the simulation code. However, it can be viewed as a bias with an effect similar to other trim disturbances. For simplicity, it was assumed that $\epsilon_0 = 0$, and the approximate local angle of attack for the missing stabilator failure for a flight condition of 9,800 feet, Mach 0.5 was set to 0.17α.

Appendix C

Simulation Details

Table C.1 gives specific parameters for the simulations shown in the figures. The failure using the BLS identification occurs at 305 seconds. All RLS simulations use $\lambda = 0.99$, $\alpha = 10.0$, and failures which occur at 310 seconds.

Table C.1: Simulation Details

Figure Number	RLS/ BLS	Step Size[1]	β Value	Integral Action	Noise Present	Failure Type[2]	Limit Type[3]	Initial Matrices[4]
1.1	RLS	1 & 5	0	no	no	lock stab	none	(a)
3.1	RLS	1	0	no	no	lock stab	none	(a)
3.2	RLS	5	0	no	no	miss stab	none	(a)
3.3	RLS	10	0,0.1	no	no	miss stab	none	(a)
4.1	RLS	1	0.1	no	no	lock stab	none	(a)
4.2	RLS	10	0.1	no	no	lock stab	none	(a)
4.3	RLS	10	0.1	no	no	lock stab	none	(a)
4.4	RLS	10	0.1	no	no	lock stab	all	(a)
4.5	RLS	10	0.1	no	no	lock stab	all	(a)
5.2	BLS	1	0.1	yes	no	lock stab	LS	(e)
5.3	RLS	1	0.1	yes	no	lock stab	LS	(a)
5.4	RLS	1	0.1	yes	no	miss stab	LS	(a)
5.5	RLS	10	0.1	yes	no	lock stab	LS	(a)
6.1	RLS	1	0.1	no	yes	lock stab	LS	(a)
6.2	RLS	10	0.1	no	yes	lock stab	LS	(a)
7.1	RLS	10	0.1	no	no	lock stab	LS	(b)
7.2	RLS	10	0.1	no	no	lock stab	LS	(c)
7.3	RLS	1	0.1	yes	no	lock stab	LS	(d)
7.4	RLS	30	0.1	no	no	lock ail	all	(a)
7.5	RLS	50	0.1	no	no	lock ail	all	(a)
7.6	RLS	10	0.1	no	no	lock st/ai	LS	(a)
7.7	RLS	30	0.1	no	no	lock st/ai	LS	(a)
8.1	BLS	10	0	no	no	lock stab	none	(a)
8.2	RLS	1	0	no	no	none	none	(a)
8.3	RLS	10	0.1	no	no	lock stab	SCI	(a)

[1] in °/s

[2] either locked left stabilator, missing left stabilator, locked left aileron, or combined locked left stabilator and locked left aileron

[3] *all* represents all four methods described in Chapter 4; *LS* represents the least-squares approximation to commanded accelerations method; *SCI* represents the scaling of control inputs method

[4] letter corresponds to the matrices listed in Table A.3

Appendix D

Gaussian Random Number Generator

The noise added to the system is modeled as Gaussian and white. Uniformly distributed random numbers between 0 and +1 are converted to a Gaussian distribution with a zero-mean and a unit variance using the Box-Muller method [21]. A uniformly distributed random number generator is written in FORTRAN and generates a random real number between 0 and +1 [20]. A seed, M, must initially be supplied by the user for the first call of the random number generator function, but subsequent calls use the seed from the previous execution of the generator. Upon calling the random number generator, $M = M \cdot 65539$. Then, if M is negative, M is recalculated as

$$M = (M + 1) + 2^N - 1, \tag{D.1}$$

where N is the number of bits defining the size of words on the particular machine that is executing the code. For this thesis, $N = 32$. The random number *rand* is calculated as

$$rand = M \cdot \frac{1}{2^N - 1}. \tag{D.2}$$

Two calls to the **rand** routine generate two uniformly distributed random variables *rand* between 0 and +1 to be passed to the Gaussian conversion routine. The numbers are first transformed to the range between −1 and +1 using the following equations:

$$\begin{aligned} v_1 &= 2.0 \cdot (\mathsf{rand} - 0.5), \\ v_2 &= 2.0 \cdot (\mathsf{rand} - 0.5), \\ r &= v_1^2 + v_2^2, \end{aligned} \tag{D.3}$$

where **rand** is a call to the uniformly distributed random number generator between 0 and +1. The r variable is the radius squared of the random point on the (v_1, v_2) plane. The r value is tested so that it is always less than 1. Then, the region uniformly covered by (v_1, v_2) is a circle and the argument for the square root is positive, as well.

The two resulting random numbers v_1 and v_2 are remapped to two Gaussian numbers using the following equations:

$$\begin{aligned} fac &= \sqrt{-2.0 \frac{ln(r)}{r}}, \\ gstore &= v_1 \cdot fac, \\ gaus &= v_2 \cdot fac. \end{aligned} \tag{D.4}$$

The value in *gstore* is saved and used during the next call to the gaussian routine. The value in *gaus* is the output of the routine. The parameter with noise is calculated as

$$param = param + \sigma \cdot gaus, \tag{D.5}$$

where *param* is any of the parameters and σ is the corresponding standard deviation shown in Table 6.1.

The same initial seed was used for all runs including noise shown in the thesis. However, simulations for other seeds were also performed, without observing significant differences.

Bibliography

[1] C. Y. Huang and R. F. Stengel, "Restructurable control using proportional-integral implicit model following," *Journal of Guidance, Control, and Dynamics*, vol. 13, no. 2, pp. 303-309, 1990.

[2] P. R. Chandler, M. Pachter, and M. Mears, "System identification for adaptive and reconfigurable control," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 3, pp. 516-524, 1995.

[3] Y. Ochi and K. Kanai, "Application of restructurable flight control system to large transport aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 2, pp. 365-370, 1995.

[4] Y. Ochi and K. Kanai, "Design of restructurable flight control systems using feedback linearization," *Journal of Guidance, Control, and Dynamics*, vol. 14, no. 5, pp. 903-911, 1991.

[5] M. R. Napolitano, S. Naylor, C. Neppach, and V. Casdorph, "On-line learning nonlinear direct neurocontrollers for restructurable control systems," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 1, pp. 170-176, 1995.

[6] J. E. Groszkiewicz, "Multivariable adaptive algorithms for reconfigurable flight control," (M.S. thesis), Carnegie Mellon University, Pittsburgh, PA, December 1994.

[7] R. W. Brumbaugh, "An aircraft model for the AIAA controls design challenge," *Proc. of the AIAA Guidance, Navigation, and Control Conference*, New Orleans, LA, 1991.

[8] Honeywell Technology Center & Lockheed Martin Skunk Works, *Multivariable Control Design Guidelines*, final report for the program: "Application of Multivariable Control Theory to Aircraft Control Laws," Minneapolis, MN, 1996.

[9] M. Bodson and J. Groszkiewicz, "Multivariable adaptive algorithms for reconfigurable flight control," *Proc. of the IEEE Conference on Decision and Control*, Orlando, FL, 1994, pp. 3330-3335.

[10] D. G. Ward and R. L. Barron, "A self-designing receding horizon optimal flight controller," *Proc. of the American Control Conference*, Seattle, WA, 1995, pp. 3490-3494.

[11] M. Bodson, "An adaptive algorithm with information-dependent data forgetting," *Proc. of the Automatic Control Conference*, Seattle, WA, June 1995, pp. 3485-3489.

[12] M. de Mathelin and M. Bodson, "Multivariable model reference adaptive control without constraints on the high-frequency gain matrix," *Automatica*, vol. 31, no. 4, pp. 597-604, 1995.

[13] W. C. Durham, "Constrained control allocation," *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 4, pp. 717-725, 1993.

[14] K. A. Bordignon and W. C. Durham, "Closed-form solutions to the constrained control allocation problem," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 5, pp. 1000-1007, 1995.

[15] W. C. Durham and K. A. Bordignon, "Multiple control effector rate limiting," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 1, pp. 30-37, 1996.

[16] M. Pachter, P. R. Chandler, and M. Mears, "Reconfigurable tracking control with saturation," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 5, pp. 1016-1022, 1995.

[17] M. Bodson and W. A. Pohlchuck, "Control allocation in reconfigurable flight control," to appear in the *Proc. of the AIAA Guidance, Navigation, and Control Conference*, New Orleans, LA, June 1997.

[18] M. Bodson and W. A. Pohlchuck, "Integral compensation in adaptive algorithms for reconfigurable flight control," *Proc. of the AIAA Guidance, Navigation, and Control Conference*, San Diego, CA, 1996.

[19] M. R. Jackson and D. F. Enns, "A concept for adaptive performance optimization on commercial transport aircraft," *NASA Contractor Report 186034*, Edwards, CA, 1995.

[20] L. Nyhoff and S. Leestma, *Fortran 77 for Engineers and Scientists*, New York: Macmillan, 1992.

[21] S. K. Park and K. W. Miller, "Random number generators: good ones are hard to find," *Communications of the ACM*, vol. 31, no. 10, pp. 1192-1201, 1988.

[22] P. M. Embree, *C Algorithms for Real-Time DSP*, Upper Saddle River, NJ: Prentice Hall, 1995.

[23] J. H. Blakelock, *Automatic Control of Aircraft and Missiles, Second Edition*, John Wiley & Sons, Inc., New York, 1991.